Causal Discovery with Metadata-Informed Latent Types

Philippe Brouillard[1,2] **Alexandre Drouin**[2,3] **Dhanya Sridhar**[1,2]

[1]Université de Montréal
[2]Mila - Quebec AI Institute
[3]ServiceNow Research

Abstract

Causal discovery seeks to recover causal structure from data but the underlying graph is typically identifiable only up to its Markov equivalence class. Yet real-world systems often exhibit redundancy, where groups of variables share similar causal roles. We introduce a Bayesian causal discovery framework that leverages variable-level metadata to infer latent types and constrain causal interactions across variables. We model causal graphs as type-consistent DAGs and propose t-DiBS, a fully differentiable method that jointly learns variable types, graph structure, and metadata representations. Our approach enables principled uncertainty quantification and integrates expressive neural models for metadata. We provide theoretical results showing that, under structured assumptions, metadata combined with typing can improve identifiability beyond classical limits. Empirically, we demonstrate improved performance over standard causal discovery methods on synthetic and pseudo-real datasets, with detailed analysis demonstrating the benefit of joint type and structure learning. These results establish metadata-driven typing as a principled approach to identifiable causal discovery.

1 INTRODUCTION

Causal reasoning lies at the core of scientific inquiry and human cognition. Despite being challenging, identifying causal relations is commonly performed by humans even in extremely low-data regimes [Kemp et al., 2010]. Humans routinely exploit prior causal knowledge to infer new causal relations: by reasoning through analogy based on similarities between entities [Zhu and Gopnik, 2023] or between causal relations themselves [Walker and Gopnik, 2014, Rett et al., 2021, 2022], and by efficiently acquiring new causal knowledge through informative interventions and well-chosen questions [Goddu and Walker, 2020, Lapidow et al., 2023]. This hints at the fact that in many real-world problems, there is a structure that can be leveraged to infer causal relations.

By contrast, in the field of causal discovery, where algorithms aim to infer causal relations from data, the systematic reuse of prior knowledge or external information remains rare. In its most general form, causal discovery is fundamentally challenging: the space of possible graphs grows super-exponentially with the number of variables, and even with infinite data, the true causal graph is generally identifiable only up to a Markov equivalence class. To address identifiability, prior work has introduced strong assumptions, such as restrictive functional forms of the causal mechanisms (e.g., Gaussian models with equal variance [Peters and Bühlmann, 2014] or additive noise models [Hoyer et al., 2008]). While these assumptions are mathematically convenient, they rarely reflect the structure of real-world systems. In this work, we seek to narrow this gap by developing a causal discovery framework that leverages metadata to exploit structures present in real-world systems.

Real-world causal discovery problems rarely correspond to the fully unconstrained setting in which any DAG is equally plausible a priori. Instead, they arise from systems shaped by physical constraints, such as biological evolution, which impose regularities on their causal structure. One such regularity, which appears ubiquitously across complex systems, is redundancy: multiple variables implement similar or overlapping causal mechanisms. Redundancy naturally emerges because it confers robustness, adaptability, and fault tolerance. This is a well-documented feature of biological, genomic, and neural systems [Edelman and Gally, 2001, Wagner, 2005, Whitacre, 2010, Sporns and Betzel, 2016], and is also reflected in network-level properties such as repeated motifs and scale-free organization [Albert et al., 2000, Albert and Barabási, 2002, Siegenfeld and Bar-Yam, 2020]. Crucially, redundancy implies that variables are not independent causal entities, but instead belong to a smaller

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:785–818.

number of latent groups that share similar causal roles. From a causal discovery perspective, this observation is central: when variables can be grouped into redundant classes with similar causal behavior, the effective complexity of the discovery and the identifiability problem are substantially reduced [Brouillard et al., 2022].

While these latent groups are not directly observed, they are often reflected in metadata (e.g., textual descriptions of variables) that capture shared function. For example, in gene regulatory network inference, metadata could be the DNA sequence itself, gene annotations (e.g., Gene Ontology [Consortium, 2004]) or even biological knowledge from the scientific literature [Poon et al., 2014, Song and Chen, 2009]. Metadata, therefore, provides a means to infer group membership and exploit this structure during causal discovery.

The idea that causal variables belong to latent types that constrain their interactions has appeared across disciplines. In cognitive science, humans are thought to learn abstract categories and their relations in order to efficiently infer causal structure [Kemp et al., 2010]. In the context of causal discovery, Brouillard et al. [2022] demonstrated that assuming known types can dramatically reduce the equivalence class of admissible causal graphs. However, this approach relied on the strong assumption that types are provided a priori; in this work, we instead propose to infer them from metadata

In this work, we relax this assumption by treating types as latent and inferring them from metadata associated with each variable. We propose a Bayesian causal discovery method that simultaneously learns variable types and the causal graph by leveraging metadata. The Bayesian formulation naturally accommodates prior knowledge and enables principled uncertainty quantification over both types and graphs. Moreover, the proposed method is fully differentiable, allowing the use of expressive models to learn representations of metadata.

Contributions:

- We propose a differentiable Bayesian causal discovery method that jointly infers latent variable types and the causal graph (Section 4).
- We show that our type-based generative process enables leveraging metadata and yields identifiability guarantees (Section 5).
- We demonstrate the value of our approach through diverse empirical studies on synthetic and pseudo-real data (Section 6).

2 BACKGROUND

In this section, we briefly present causal models, causal discovery and Bayesian causal discovery.

Causal Bayesian network. A causal Bayesian network (CBN) is constituted of a random vector $\boldsymbol{X} = (X_1, \cdots, X_d)$ and directed acyclic graph (DAG) $G = (V, E)$ where $|V| = d$ [Peters et al., 2017]. Furthermore, the joint distribution of $\boldsymbol{X}$ is Markov to the DAG G leading to the following factorization of the joint:

$$p(\boldsymbol{X}) = \prod_{i=1}^{d} p(X_i \mid \mathrm{pa}_i^G) \tag{1}$$

where pa_i^G are the random variables that are parents of X_i in the DAG G. The directed edges in G have a causal semantics: an edge $X_j \to X_i$ indicates that X_j is a direct cause of X_i, and the conditional distribution represents the causal mechanism by which X_i is generated from its parents.

Causal discovery. The task of causal discovery consists of learning a DAG $\hat{G}$ directly from a dataset $\mathcal{D}$. Causal discovery is a particularly challenging task since, even with infinite data, the graph is only recoverable up to its Markov Equivalence Class (MEC). Two DAGs belong to the same MEC if they encode the same set of conditional independence relations. Equivalently, they share the same skeleton (underlying undirected graph) and the same v-structures (colliders of the form $X \to Z \leftarrow Y$ where X and Y are not adjacent) [Verma and Pearl, 1990]. As in our proposed framework, additional assumptions on the generative process can alleviate the identifiability problem by shrinking the equivalence class.

Causal discovery methods are often broadly described in two main categories: constraint-based methods that rely on conditional independence tests and score-based methods that search for the DAG maximizing a score such as the likelihood. The method that we propose falls into the category of score-based methods and adopts a Bayesian formulation, enabling joint inference over graph structure, latent types, and model parameters.

Bayesian causal discovery. Compared to classical point estimate methods, Bayesian causal discovery [Friedman and Koller, 2003] is an approach that can naturally deal with uncertainty by estimating the posterior distribution over DAGs which has the form:

$$p(\boldsymbol{G} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{G}) p(\boldsymbol{G}). \tag{2}$$

This allows us to estimate expectations of the form:

$$\mathbb{E}_{p(\boldsymbol{G} \mid \mathcal{D})}[f(\boldsymbol{G})] \tag{3}$$

where f can be any function of interest. If $f(\boldsymbol{G})$ is the likelihood $p(\mathcal{D} \mid \boldsymbol{G})$, then this amounts to Bayesian model averaging.

More generally, this framework naturally extends to latent-variable settings, where one seeks joint posterior inference

over graph structure and additional unobserved quantities. In our setting, this corresponds to jointly inferring the causal graph and latent variable types from observational data and metadata.

3 SETTING

In this section, we explicitly present our typing assumption and the generative model that we assume.

3.1 TYPING ASSUMPTIONS

Motivated by the observation that real-world systems often exhibit redundancy where multiple variables share similar causal roles, we assume that causal variables can be grouped into a finite number of latent types. Variables of the same type are assumed to participate in causal relations in a consistent manner, inducing structural regularities in the underlying causal graph.

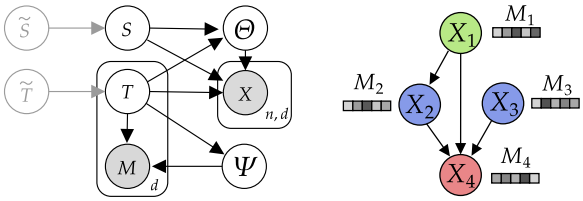
Formally, we consider directed acyclic graphs augmented with types. Each variable X_i is assigned a type $t_i \in \mathcal{T}$, and causal relations are constrained at the level of types: for any pair of distinct types $t_i, t_j \in \mathcal{T}$, causal edges may exist in only one direction between variables of these types. This induces a t-DAG, in which the directionality between types is shared across all variables belonging to those types. We define more formally t-DAGs in Def 1.

Definition 1 (t-DAG)**.** *A* t-DAG *is a directed acyclic graph* $G = (V, E)$ *together with a type assignment* $T : V \to \mathcal{T}$*, where* $|\mathcal{T}| = k$*, satisfying the following properties:*

1. *(*Type consistency*) For any two distinct types* $t_i, t_j \in \mathcal{T}$*, all edges between variables of type* t_i *and variables of type* t_j *have the same direction.*
2. *(*No intra-type edges*) If* $T(u) = T(v)$*, then neither* (u, v) *nor* (v, u) *belongs to* E*.*
3. *(*Acyclic type graph*) The induced graph on types is acyclic. This graph is defined by an edge* $t_i \to t_j$ *whenever there exists* $u \to v$ *with* $T(u) = t_i$ *and* $T(v) = t_j$*.*

A similar type-consistency assumption was previously shown to substantially reduce the size of the equivalence class and can lead to identifiability when types are known [Brouillard et al., 2022]. In our setting, however, types are latent and inferred jointly with the causal graph using variable-level metadata, allowing the model to discover redundant structure directly from data.

In Figure 1.b we show an example of a t-DAG where the different colors represent the type of the variable. It respects type consistency since the orientation between distinct types is the same: both X_2, X_3 are causes of X_4. Furthermore, metadata is associated to each variable as we will describe in the next section.



(a) Data generating process (b) t-DAG with metadata

Figure 1: In a), generative model plate diagram with observed variables $\boldsymbol{X}$, $\boldsymbol{M}$ (shaded) and latent variables S, T, Θ, Ψ. In b), example t-DAG where node colors indicate latent types and M_i denotes the metadata of X_i.

3.2 GENERATIVE MODEL

We assume a generative model that reflects our typing assumption. We assume the following factorization of the joint distribution $p(\boldsymbol{M}, \boldsymbol{X}, \boldsymbol{S}, \boldsymbol{T}, \boldsymbol{\Theta}, \boldsymbol{\Psi})$ (see Figure 1.a):

$$p(\cdot) = p(\boldsymbol{S})p(\boldsymbol{T})p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S}, \boldsymbol{T}) \cdot p(\boldsymbol{M} \mid \boldsymbol{T}, \boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T}, \boldsymbol{\Theta}) \tag{4}$$

where:

$$p(\boldsymbol{M} \mid \boldsymbol{T}, \boldsymbol{\Psi}) = \prod_{i=1}^{d} p(M_i \mid T_i, \boldsymbol{\Psi}) \tag{5}$$

$$p(\boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T}, \boldsymbol{\Theta}) = \prod_{i=1}^{n} \prod_{j=1}^{d} p\left(X_j^{(i)} \mid \boldsymbol{S}, \boldsymbol{T}, \boldsymbol{\Theta}\right). \tag{6}$$

The observed variables are $\boldsymbol{X} \in \mathbb{R}^d$, the causal variables, and $\boldsymbol{M} \in \mathbb{R}^{d \times m}$, the metadata related to each variable. The other variables are $\boldsymbol{\Theta}$, the parameters defining the relations between the different causal variables, $\boldsymbol{\Psi}$, the parameters defining the functional form of the metadata, and $\boldsymbol{S}$, $\boldsymbol{T}$ refers to the skeleton of the graph and the types of each variable. Together, $\boldsymbol{S}$ and $\boldsymbol{T}$ fully specify the DAG $\boldsymbol{G}$ between the variables $\boldsymbol{X}$ and encode the typing assumption. In Figure 1.b, we show an example of a dataset $\mathcal{D} = \{((x_i^{(j)})_{j=1}^n, m_i)\}_{i=1}^d$ generated by this process that contain metadata.

More precisely, $\boldsymbol{G}$ has a factorization, in terms of $\boldsymbol{S}$ and $\boldsymbol{T}$, that enforce type-consistency:

$$\boldsymbol{G} := \boldsymbol{S} \odot (\boldsymbol{T}\boldsymbol{U}\boldsymbol{T}') \tag{7}$$

where $\odot$ is the elementwise product, the symmetric matrix $\boldsymbol{S} \in \{0, 1\}^{d \times d}$ represents the skeleton of the graph, the matrix $\boldsymbol{U} \in \{0, 1\}^{k \times k}$ is a strictly upper triangular matrix and the matrix $\boldsymbol{T} \in \{0, 1\}^{d \times k}$ represent the assignment of each node to a type: $T_{ij} = 1 \iff$ the type of X_i is t_j. In a sense, $\boldsymbol{T}$ encodes both the types and the causal order among

types. This parametrization enforces type-consistency by construction: $\boldsymbol{U}$ is strictly upper triangular, which defines an acyclic ordering over types and thus, for any pair of distinct types t_a, t_b, edges may only exist in one direction between them.

The likelihood model is a CBN given by:

$$p(\boldsymbol{X} \mid \boldsymbol{G}, \boldsymbol{\Theta}) = \prod_{i=1}^{d} p_i(X_i \mid \mathrm{pa}_i^{\boldsymbol{G}}; \boldsymbol{\Theta}). \tag{8}$$

In our experiments, each conditional p_i follows a normal distribution with the parameters given by $\mathrm{NN}_i(\boldsymbol{X} \odot \boldsymbol{G})$ where NN is a neural network parameterized by $\boldsymbol{\Theta}_i$.

Finally, the metadata follows a distribution that depends on its type and is parametrized by $\boldsymbol{\Psi}$, i.e., $\boldsymbol{M}_i \sim P(\boldsymbol{M}_i \mid T_i, \boldsymbol{\Psi})$. In the synthetic experiments, we assume the generative model of metadata to be a transformation of a Gaussian mixture model (GMM):

$$p(\boldsymbol{M} \mid \boldsymbol{T}, \boldsymbol{\Psi}) = \prod_{i=1}^{d} g_{\boldsymbol{\Psi}}(\mathcal{N}(\boldsymbol{\mu}_{T_i}, \boldsymbol{\Sigma}_{T_i})) \tag{9}$$

where $\boldsymbol{\mu} \in \mathbb{R}^m$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$. Finally, the types follow $\boldsymbol{T} \sim Cat(\boldsymbol{\pi})$ where $\boldsymbol{\pi}$ is a vector of probability of each type (i.e., $\sum_i \pi_i = 1$) and the skeleton $\boldsymbol{S}$ is defined as $\boldsymbol{S} := \hat{\boldsymbol{S}}^T + \hat{\boldsymbol{S}}$ where $\hat{S}_{ij} \sim \mathrm{Bern}(p), \forall i > j$.

4 METHOD

Our goal is to recover the causal graph $\boldsymbol{G}$ from observational data $\boldsymbol{X}$ and variable-level metadata $\boldsymbol{M}$. Under our formulation, this amounts to jointly inferring the graph skeleton $\boldsymbol{S}$ and the type assignments $\boldsymbol{T}$, since together they fully specify $\boldsymbol{G}$. To this end, we introduce t-DiBS, a differentiable Bayesian causal discovery method that jointly infers causal structure and latent variable types from data and metadata. The method builds on the differentiable Bayesian structure learning framework DiBS from Lorch et al. [2021], but extends it to support typed causal graphs and metadata-driven inference.

4.1 PARAMETRIZATION

To enable differentiable inference over discrete graph structure and variable types, we follow Lorch et al. [2021] and Hägele et al. [2023] and introduce continuous latent variables $\tilde{\boldsymbol{S}}$ and $\tilde{\boldsymbol{T}}$ that parameterize, respectively, the graph skeleton $\boldsymbol{S}$ and type assignments $\boldsymbol{T}$. This continuous relaxation enables a fully differentiable variational approach for joint inference over graph structure, latent types, and model parameters. We describe below the resulting generative parametrization. In the following section, we present a framework which allows us to approximate the posterior using Stein Variational Gradient Descent (SVGD) [Liu and Wang, 2016].

The complete generative model is represented in Figure 1.a (with the continuous latent variables in grey) and leads to the following factorization of $p(\boldsymbol{M}, \boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{S}, \boldsymbol{T}, \tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}})$:

$$\begin{aligned} p(\cdot) = {} & p(\tilde{\boldsymbol{S}})p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}})p(\tilde{\boldsymbol{T}})p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})p(\boldsymbol{\Psi} \mid \boldsymbol{T}) \\ & \cdot p(\boldsymbol{M} \mid \boldsymbol{T}, \boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta}, \boldsymbol{S}, \boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S}, \boldsymbol{T}). \end{aligned} \tag{10}$$

Generative model of the skeleton $\boldsymbol{S}$:

$$p_\alpha(\boldsymbol{S} \mid \tilde{\boldsymbol{S}}) = \prod_{i<j} \mathrm{Bern}\Big(S_{ij}; \sigma_\alpha(\tilde{S}_{ij})\Big), \tag{11}$$

where $\sigma_\alpha(\cdot)$ is the sigmoid function with inverse temperature α. The prior $p(\tilde{\boldsymbol{S}})$ is:

$$p(\tilde{\boldsymbol{S}}) \propto \prod_{i<j} \underbrace{\mathcal{N}(\tilde{S}_{ij}; 0, \sigma_{\tilde{S}}^2)}_{\text{inference stability}}. \tag{12}$$

Finally, a sparsity prior can be used, for example, for Erdős-Rényi graph we can use $p(\boldsymbol{S}) \propto q^b(1-q)^{\binom{d}{2}-b}$ where q is the probability of adding an edge to the skeleton and $b := \sum_{i<j} S_{ij}$.

Generative model of types:

$$p_\gamma(\boldsymbol{T} \mid \tilde{\boldsymbol{T}}) = \prod_{i=1}^{d} p_\gamma(T_i \mid \tilde{\boldsymbol{T}}_i) \tag{13}$$

where $\tilde{\boldsymbol{T}}_i \in \mathbb{R}^k$ and $T_i \mid \tilde{\boldsymbol{T}}_i \sim Cat(\mathrm{softmax}_\gamma(\tilde{\boldsymbol{T}}))$ with γ as the inverse of the temperature of the softmax. As our type prior, we use a Dirichlet prior to encourage sharp one-hots:

$$p(\tilde{\boldsymbol{T}}) \propto \prod_{i=1}^{d} \underbrace{\mathrm{Dir}(\eta_i;\, \boldsymbol{\zeta})}_{\text{sharp onehot}} \prod_{j=1}^{k} \underbrace{\mathcal{N}(\tilde{\boldsymbol{T}}_{ij} \mid 0, \sigma_{\tilde{T}}^2)}_{\text{inference stability}}, \tag{14}$$

where $\eta_i = \mathrm{softmax}_\gamma(\tilde{\boldsymbol{T}}_i)$ and every $\zeta_i < 1$ corresponds to having a probability distribution where most of the mass is around the vertices of the simplex.

4.2 INFERENCE

In this section, we describe how posterior inference over discrete structure (graph and types) can be carried out via our continuous relaxation. In Proposition 1, we show that expectations under the discrete posterior can be equivalently expressed as expectations over the relaxed continuous posterior. The derivation is provided in Appendix A.1.

Proposition 1. *Following the proposed generative model, the expectation* $\mathbb{E}_{p(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi}\mid\mathcal{D})}[f(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})]$ *with discrete variable is equal to:*

$$\mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}\mid\mathcal{D})}\left[\frac{\mathbb{E}_{p(\boldsymbol{S}\mid\tilde{\boldsymbol{S}}),p(\boldsymbol{T}\mid\tilde{\boldsymbol{T}})}[w f(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})]}{\mathbb{E}_{p(\boldsymbol{S}\mid\tilde{\boldsymbol{S}}),p(\boldsymbol{T}\mid\tilde{\boldsymbol{T}})}[w]}\right] \tag{15}$$

where $w := p(\boldsymbol{X} \mid \boldsymbol{\Theta}, \boldsymbol{S}, \boldsymbol{T}) p(\boldsymbol{M} \mid \boldsymbol{T}, \boldsymbol{\Psi}) p(\boldsymbol{\Theta} \mid \boldsymbol{S}, \boldsymbol{T}) p(\boldsymbol{\Psi} \mid \boldsymbol{T})$.

We perform posterior inference of $p(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \mathcal{D})$ using Stein Variational Gradient Descent (SVGD) [Liu and Wang, 2016], which approximates the posterior with a set of interacting particles. Each particle is driven toward high-density regions of the posterior by the score function $\nabla \log p(\cdot)$, while a kernel-based repulsive term encourages diversity among particles. See Appendix B.1 for a more detailed explanation of this method and Appendix B.2 for our complete algorithm. In proposition 2, we provide a tractable expression for one of the scores used by SVGD under our generative model. See Appendix A.2 for the derivation and the scores of $\tilde{\boldsymbol{T}}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\Psi}$.

Proposition 2. *The score of the log-posterior* $\nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \mathcal{D})$ *is equal to:*

$$\nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}) + \frac{\nabla_{\tilde{\boldsymbol{S}}} \mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[w]}{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[w]} \tag{16}$$

with w *as defined in Proposition 1.*

To estimate the numerator of Eq. 16, which takes the gradient w.r.t. the same parameters of the expectation, one can estimate by the Monte Carlo method using Reinforce [Williams, 1992] or the Gumbel-softmax estimator [Jang et al., 2017, Maddison et al., 2017]. We show in Appendix A.4 the derivation for both of these methods.

Annealing. Both for the skeleton and the types, the inverse temperatures α and γ are gradually increased during training progressively concentrating probability mass on discrete structures. In the limit $\alpha, \gamma \to \infty$, the sigmoid and softmax converge to hard thresholding and one-hot assignments, respectively, recovering discrete graph skeletons and type labels while allowing smooth optimization throughout inference. In Proposition 3, we show how expectations under the relaxed posterior converge to expectations under the discrete posterior.

Proposition 3. *Let* α *and* γ *denote the inverse temperatures of the Bernoulli and categorical relaxations for the skeleton and types, respectively. As* $\alpha, \gamma \to \infty$*, we have that*

$$\mathbb{E}_{p(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \mathcal{D})}[f(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{\Theta}, \boldsymbol{\Psi})] \to$$
$$\mathbb{E}_{p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \mathcal{D})}[f(\boldsymbol{S}_\infty(\tilde{\boldsymbol{S}}), \boldsymbol{T}_\infty(\tilde{\boldsymbol{T}}), \boldsymbol{\Theta}, \boldsymbol{\Psi})]$$

where $\boldsymbol{S}_\infty(\tilde{\boldsymbol{S}})_{ij} := \mathbf{1}_{\tilde{\boldsymbol{S}}_{ij} > 0}$ *and* $\boldsymbol{T}_\infty(\tilde{\boldsymbol{T}})_i := \text{onehot}\left(\arg\max_j \tilde{\boldsymbol{T}}_{ij}\right)$.

5 THEORETICAL RESULTS

In this section, we characterize the class of t-DAGs, showing that it is a restricted space of DAGs, and we prove that metadata under the typing assumption can lead to the identifiability of the causal graph.

5.1 CHARACTERIZATION OF T-DAGS

We begin by characterizing the class of graphs induced by the typing assumption. Intuitively, assigning variables to k latent types imposes a global ordering over types, which in turn bounds the length of any causal path.

Formally, Proposition 4 shows that the space of t-DAGs with k types is equivalent to the union of DAGs with height at most k (the height of a DAG is the length of its longest directed path). In other words, rather than allowing arbitrarily long causal chains, typing restricts graphs to at most k levels of causal depth.

Proposition 4 (Equivalence to bounded-height DAGs)**.** *Let* $\mathcal{G}_d(\ell)$ *and* $\mathcal{G}_d^k$ *be the sets of all DAGs of height* ℓ *and t-DAGs with* d *vertices and* k *types. Then,* $\mathcal{G}_d^k = \bigcup_{\ell=0}^{k-1} \mathcal{G}_d(\ell)$.

Proposition 5 further shows that, for fixed k, the fraction of such t-DAGs among all DAGs converges to zero as the number of variables grows. In other words, the typing assumption drastically shrinks the hypothesis space in high dimensions, providing intuition for why it can lead to improved identifiability.

Proposition 5 (Limits of the cardinality, t-DAG vs DAG)**.** *The ratio* $\frac{|\mathcal{G}_d^k|}{|\mathcal{G}_d|}$ *converges to* 0 *when* $d \to \infty$ *and* k *is fixed.*

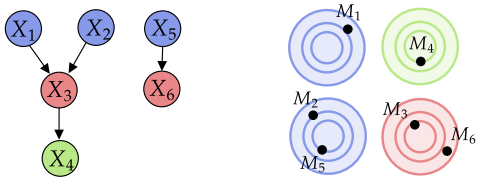
We also performed empirical validation in Appendix C.3 and give additional comparison to a similar class of low-rank DAGs from Lopez et al. [2022].

If we assume that the types are known, then the typing assumption leads to an equivalence class that is much smaller than the classical Markov Equivalence Class [Brouillard et al., 2022]. As an example, in Fig 2.a, the t-DAG is identifiable: the left component is orientable from its v-structure and Meek's laws [Meek, 1995] and, since the types are known, the right component is also orientable. However, when the types are unknown, this result does not apply, and we cannot identify the graph. We now show that metadata provides the additional signal needed to recover both types and causal structure.

5.2 IDENTIFIABILITY OF THE T-DAG

In this section, we show that the combination of typing and metadata can improve identifiability beyond classical Markov equivalence. While prior work assumed types to be known [Brouillard et al., 2022], we establish that latent types inferred from metadata are sufficient to recover the full causal graph in large systems.

To study this setting, we consider a random sequence of growing t-DAGs defined in Def 2 (similar to the process from Brouillard et al. [2022]), which models the incremental



(a) t-DAG (b) Metadata

Figure 2: In a), an example of a t-DAG that is only identifiable when type information is incorporated. In b), an example of metadata distributions where information from the MEC is necessary to correctly infer the underlying types.

addition of variables while preserving the underlying type structure.

Definition 2 (Random sequence of growing t-DAG)**.** *We define a random sequence of t-DAGs* $(G_d^{T_d})_{d=0}^{\infty}$ *with* $G_d^{T_d} = (V_d, E_d)$ *and* $|V_d| = d$*, such that* $G_0^{T_0} = (\emptyset, \emptyset)$*. Each new t-DAG* $G_d^{T_d}$ *in the sequence is obtained from* $G_{d-1}^{T_{d-1}}$ *as follows: Create a new vertex* v_d *and sample its type* t *from a categorical distribution with probabilities* $p_1, ..., p_k$*. Let* $V_d = V_{d-1} \bigcup \{v_d\}$*. Let* $T_d(v_d) = t$ *and* $T_d(v_j) = T_{d-1}(v_j), \forall v_j \in V^{d-1}$*; To obtain* E_d*, for every vertex* $v_i \in V_{d-1}$*, add the edge* (v_i, v_d) *to* E_{d-1} *with probability* $p_{T_d(v_i), T_d(v_d)}$*.*

As the t-DAGs considered grow larger, the probability of them containing full-length oriented path in their MEC grows exponentially fast.

Theorem 1 (Probability of oriented full-length path). *Let* $(G_d^{T_d})_{d \geq 0}$ *be a random sequence of growing t-DAGs with* k *types, type DAG of height* ℓ*, with* $p_i \in (0, 1)$ *and* $p_{ij} \in (0, 1)$*. Assume the type DAG has a single component. Let* C *be the indicator that the MEC of* $G_d^{T_d}$ *contains an oriented path of length* ℓ*. Then as* d *increases:*

$$P(C = 0 \mid d) \leq O(d) \cdot e^{-dr} \tag{17}$$

where $r = -\frac{1}{\ell} \ln(\alpha^*) > 0$ *and* $\alpha^* \in (0, 1)$ *depends only on* $\{p_i\}$ *and* $\{p_{ij}\}$ *(defined in the proof).*

Intuitively, Theorem 1 shows that as the graph grows, the Markov equivalence class contains long oriented paths that reveal the ordering between types with high probability. In particular, $P(C = 1) \to 1$ exponentially fast as $d \to \infty$. These paths provide anchor points that allow types to be inferred from metadata. Once types are identified on these oriented paths, the learned metadata-to-type mapping generalizes to the remaining variables, enabling recovery of the full type assignment and, in turn, orientation of all inter-type edges.

Theorem 2 (Identifiability with metadata). *Assume that (i) the data-generating graph is a t-DAG with fixed* k *types, (ii) the metadata are i.i.d. conditioned on type and uniquely identify types, and (iii) the Markov equivalence class is recovered. Then, with probability approaching one as* $d \to \infty$*, the causal graph* G *is identifiable from the joint distribution* $p(\boldsymbol{X}, \boldsymbol{M})$*.*

Proof sketch. The proof proceeds in three steps, exploiting the interplay between the graph structure and the metadata.

- Since we assumed that the MEC is recovered, under the growing t-DAG sequence, the MEC will include oriented paths of length ℓ as $d \to \infty$ by Thm 1, yielding anchor vertices at each of the ℓ heights.
- Assuming metadata are i.i.d. per type, these anchors provide labeled samples whose number grows with d. Standard learning arguments then allow learning a metadata-to-layer classifier with vanishing error.
- Applying this classifier to all vertices assigns each to a layer. By type-consistency, all inter-layer edges are oriented according to the layer ordering, yielding recovery of the full graph G.

In Appendix D, we provide the proofs and also empirical validation of these results.

We note that, in parallel to our identifiability result, under some stronger assumptions on the distribution of the metadata, other existing identifiability results can be applied. Namely, in Appendix D.3, we discuss how metadata generated by a GMM or a GMM transformed by piecewise linear functions can be identified.

5.3 PARTIAL IDENTIFIABILITY UNDER METADATA COARSENING

In practice, metadata may be informative without uniquely identifying types. For example, two types might have metadata coming from the same distribution. Formally, we define an equivalence relation on types: $t_i \sim t_j$ if their metadata distributions are indistinguishable. The recoverable structure is then captured by the *quotient DAG*: the graph obtained by collapsing each equivalence class to a single node and inheriting the edges between classes.

Proposition 6 (Partial identifiability under coarsening)**.** *Identifiability holds at the quotient level: edges between variables whose aggregate types are distinguishable are oriented beyond the MEC, while edges within an aggregate class are oriented only to the extent permitted by* v*-structures. The quotient is well-defined as a DAG only when no type outside an equivalence class lies on a directed path between two types within it.*

We provide a proof in Appendix D.1.3. In the extreme case where all types are indistinguishable, identifiability reverts to that of the MEC. An empirical validation in Appendix F.4 confirms that t-DiBS degrades gracefully when metadata

is completely uninformative rather than failing catastrophically.

6 EXPERIMENTS

We compare the performance of t-DiBS to several different causal discovery methods that do not leverage metadata information. We first evaluate on a wide range of synthetic datasets, then conduct a targeted analysis of metadata's contribution to joint inference. Finally, we consider a pseudo-real task where types and metadata are well defined. The code for t-DiBS is available on this Github repo.

6.1 SYNTHETIC DATASETS

We first compare t-DiBS to t-DiBS without metadata (t-DiBS--), the type-agnostic algorithm DiBS [Lorch et al., 2021], PC [Spirtes et al., 2000], GSP [Solus et al., 2021], and GES [Chickering, 2002] on synthetic datasets with $d = 20$ variables, $k = \{2, 5\}$ types, and either linear or nonlinear mechanisms. We also included $d = 50$ variables for linear mechanisms. To generate these datasets, we followed the generating process from Section 3 and Def 2 (see Appendix E.1) and generated $n = 500$ samples per dataset. We report in Appendix F.2 and G.1 the hyperparameters used for t-DiBS and the baselines, respectively. For t-DiBS, most hyperparameters follow the original DiBS work. We also detail the baselines' implementation, including the nonparametric DAG bootstrap used to obtain a likelihood from PC, GSP, and GES's output graphs. In Fig 3 and 4, we report for ($d = 50$, $k = 5$, linear) and ($d = 20$, $k = 2$, nonlinear) the expected Structural Hamming Distance ($\mathbb{E}$-SHD) and the negative log-likelihood on unseen interventional distribution (I-NLL); see Appendix F.1 for an explanation of these metrics and Appendix F.3 for the full results. Overall, t-DiBS achieves the best performance. While t-DiBS without metadata often has a good performance, t-DiBS with metadata is slightly better, showing that the performance is not solely explained by the structure induced by the typing assumption. In Appendix F.5, we also compare the performance to the method typed-PC [Brouillard et al., 2022] that has access to the ground-truth information. Overall, the gap between t-DiBS and this privileged method remains small.

6.2 DETAILED ANALYSIS

In the following section, we inspect more closely some properties of learning types from metadata. Specifically, we show that (i) metadata-driven typing is particularly beneficial when the vanilla MEC is large, but the typed MEC is small, (ii) jointly learning types and causal structure outperforms two-stage approaches based on metadata clustering followed by causal discovery, and (iii) learned type representations enable effective transfer across related graphs.

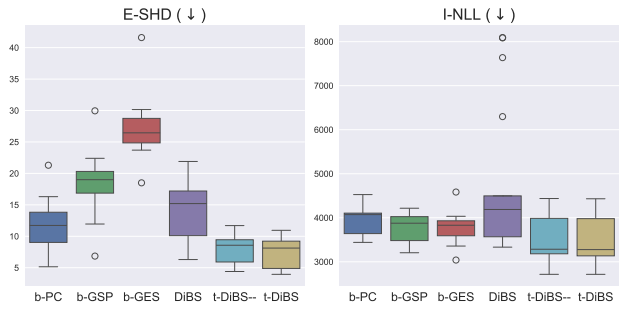


Figure 3: Performance of different causal discovery methods on 10 synthetic datasets with 20 variables, 2 types, and nonlinear mechanisms.

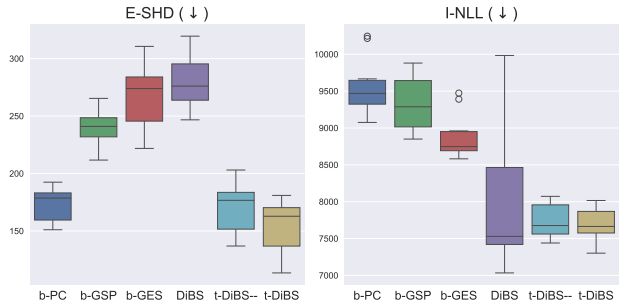


Figure 4: Performance of different causal discovery methods on 10 synthetic datasets with 50 variables, 5 types, and linear mechanisms.

We report the expected number of reversed edges, as this metric is more informative than SHD in our setting: SHD is strongly affected by sparsity regularization and can therefore obscure differences in edge orientation performance.

Leveraging Metadata for Identifiability. While the experiments of Section 6 show that using metadata can be advantageous for recovering the causal graph, we more explicitly look at this by comparing performance on t-DAGs where the difference between the size of the MEC and the t-MEC is large. When graphs are dense, both the MEC and the t-MEC are small, since the large number of v-structures strongly constrains edge orientations.

The left panel of Figure 5 compares DiBS and t-DiBS on t-DAGs that are not identifiable without metadata. As expected, t-DiBS with metadata significantly outperforms t-DiBS without metadata. This confirms that incorporating type information effectively reduces ambiguity and can improve causal recovery.

Joint Learning of Types and Graph Structure is Advantageous. In this second study, we demonstrate that jointly learning types and the causal graph is superior to a two-stage approach in which clustering is applied first to the metadata, followed by causal discovery.

We consider a synthetic dataset with three latent types,

where the metadata are generated from a mixture of four Gaussians (instead of three). Thus, one of the types has two corresponding components as shown in Fig 2.b. A clustering algorithm applied to the metadata alone would successfully recover the four Gaussian components, but cannot, by any means, determine which components correspond to the same type. However, by using the information in the MEC, we can recover the mapping of metadata to types.

As shown in the middle panel of Figure 5, t-DiBS correctly recovers the underlying types and achieves a substantially lower SHD. This illustrates that learning types jointly with the causal graph provides additional structural constraints that cannot be recovered by clustering alone. Overall, this experiment highlights the synergy between type inference and causal discovery in t-DiBS.

Transfer Learning via Type Representations. In the final study, we investigate how learning types in t-DiBS facilitates transfer learning across graphs. We first train t-DiBS on a t-DAG. We then apply the model to a new t-DAG that shares the same mapping from metadata to types but contains no v-structures. Crucially, we reuse the learned parameter $\mathbf{\Psi}$ while re-learning the causal graph.

As shown in the right panel of Figure 5, the pretrained model achieves significantly fewer reversed edges (close to zero) compared to training t-DiBS from scratch. A closer inspection reveals that, without pretraining, t-DiBS correctly recovers the types but fails to orient edges between them. In contrast, the pretrained model successfully transfers the learned type structure, enabling accurate causal recovery even in the absence of strong structural cues. This result highlights the benefit of disentangling type learning from graph learning and demonstrates the potential of t-DiBS for transfer across related causal systems.

Taken together, these experiments demonstrate that metadata-driven typing consistently improves causal recovery and that joint inference is key to realizing this benefit.

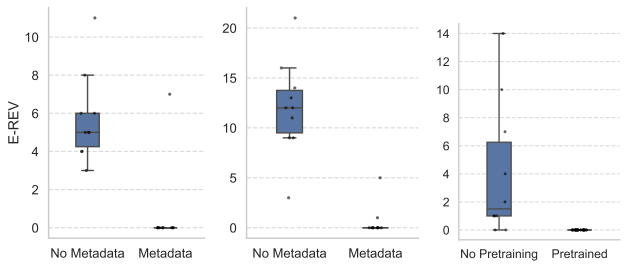


Figure 5: Experiments reporting the number of reversed edges. **Left:** identifiability gain from metadata on sparse graphs. **Middle:** joint learning versus two-stage clustering. **Right:** transfer learning via pretrained type representations.

6.3 PSEUDO-REAL DATASET

In this section, we analyze the performance of t-DiBS on a dataset generated by the neuropathic pain simulator from Tu et al. [2019]: a simulator that was fine-tuned to a real-world dataset from Zhang et al. [2016]. The generated dataset contains 222 binary variables forming a graph with 770 edges. The graph structure is particularly interesting in light of the typing assumption, as it can be partitioned into three types: pathophysiologies, patterns, and symptoms (See Fig. 2 of Tu et al. [2019] (27 pathophysiologies, 52 patterns, 143 symptoms)). As metadata, we use embeddings from a language model of the textual description of each variable (e.g., "Injury at the transition of the skull and neck"). Since our implementation operates on continuous data, we generated linear data using the graph structure (and the metadata) of Tu et al. [2019]. In Appendix E.2, we present the codebook used for the metadata and provide all experimental details.

Method	**F1** (↑)
t-DiBS	$\mathbf{0.523 \pm 0.032}$
t-DiBS-- (no metadata)	0.486 ± 0.034
PC	0.210 ± 0.012
GSP	0.030 ± 0.006

Table 1: Performance on the pseudo-real datasets.

In Table 1, we report the F1 score averaged over 10 instantiations of the dataset. The two t-DiBS variants substantially outperform PC and GSP. More importantly, t-DiBS has a better performance than t-DiBS-- (no metadata), demonstrating the benefit of leveraging metadata. The modest gap between the two methods suggests that the graph structure is already largely identifiable from the modular typing alone, with metadata providing an additional but incremental gain. GES was excluded due to its long compute time and DiBS due to issues with the acyclicity constraint (See App F.2).

Additionally, Tu et al. [2023] applied an LLM-based causal discovery approach to a subset of the causal edges in this dataset, achieving an F1 score of 0.21, well below t-DiBS (0.523). This shows that, at least for this dataset, integrating metadata inference directly into structure learning is more effective than relying on language models alone.

Overall, this experiment demonstrates that t-DiBS generalizes beyond GMM-generated metadata to realistic textual embeddings, and that our typing assumptions arise naturally in real-world structured domains.

7 RELATED WORK

Typed causal discovery. The most directly related work is Brouillard et al. [2022], which introduced the typing assumption for causal discovery and showed that known types can dramatically reduce the Markov equivalence class.

Our work builds on this foundation but relaxes the strong assumption that types are provided a priori: we instead treat types as latent and infer them jointly with the causal graph using variable-level metadata.

Latent variable groupings in graphical models. Another group of works considers settings closely related to our notion of types, where variables are grouped into latent classes that influence their probabilistic dependencies. Early examples include the hierarchical Bayesian approaches of Mansinghka et al. [2006], Kemp et al. [2006], and Kemp et al. [2010], which jointly infer class assignments and graph structure using MCMC-based inference. While conceptually related, these methods do not leverage external metadata, and their reliance on sampling limits scalability to medium-scale graphs. More distantly related are works that exploit known parameter symmetries or groupings in Gaussian graphical models to improve estimation efficiency [Højsgaard and Lauritzen, 2008, Wu and Drton, 2023, Makam et al., 2023, Boege et al., 2024], though these assume linear mechanisms and a known grouping structure.

Latent causal variable models. A related but distinct line of work models latent variables as unobserved causal nodes added to the graph. RLCD [Dong et al., 2024] and its nonlinear extension [Prashant et al., 2025] recover such latent nodes from observational data by exploiting rank deficiency of the covariance matrix, requiring each latent variable to have at least two pure observed children and a strict layered hierarchy between latent nodes. This is fundamentally different from our setting: in t-DiBS, types are structural constraints on observed variables rather than additional graph nodes, and type-consistency restricts edge directions between groups of observed variables without inducing the information bottleneck that underlies rank deficiency.

Low-rank and global structural assumptions. Other approaches focus on global structural properties of real-world graphs, such as scale-free or hub-dominated architectures. These graphs admit low-rank representations of the adjacency matrix, which have been exploited to improve scalability and sample efficiency [Lopez et al., 2022, Fang et al., 2023, Dong and Sebag, 2022]. While effective in large dimensions, such assumptions encode global regularities rather than semantic or type-based constraints tied to individual variables.

Probabilistic relational models. Probabilistic Relational Models and their causal extensions assume the existence of multiple entity types and describe causal relations at the level of a relational template graph [Maier et al., 2013, Salimi et al., 2020]. Our setting can be viewed as a degenerate case with a single entity type; however, unlike relational models, our goal is to infer variable types from metadata rather than assuming them.

Language models for causal discovery. Recent work has explored the use of large language models to perform causal discovery using textual descriptions of causal variables [Choi et al., 2022, Ban et al., 2023, Kıcıman et al., 2024, Long et al., 2023, Abdulaal et al., 2024, Busch et al., 2025], which instantiates a specific form of metadata, namely unstructured text. A key distinction is that in these methods, semantic information is treated as a static prior and is not revised in light of the learned causal structure, whereas our method jointly infers latent variable types and graph structure from data and metadata simultaneously. As a result, LLM-based approaches cannot exploit feedback between the emerging causal structure and the grouping of variables. Furthermore, these methods typically do not provide identifiability guarantees, whereas our framework comes with theoretical guarantees on graph recovery.

8 DISCUSSION

We proposed t-DiBS, a Bayesian causal discovery framework that leverages metadata to infer latent variable types and constrain causal structure. By explicitly modeling redundancy through typing, our approach substantially reduces the effective hypothesis space and can improve identifiability beyond classical Markov equivalence. We provide theoretical guarantees showing that, under some assumptions on metadata, the true causal graph becomes identifiable in large systems, and empirically demonstrate consistent gains over type-agnostic methods.

Our experiments highlight a genuine complementarity between structure and metadata: causal constraints help disambiguate metadata groupings, while metadata helps orient otherwise ambiguous edges. This bidirectional interaction underscores the importance of joint inference. The transfer learning result further suggests that learned type representations encode reusable causal knowledge that generalizes across systems sharing the same underlying type structure.

While t-DiBS degrades gracefully when metadata is uninformative (reverting to MEC-level identifiability as shown in Section 6.2), misleading metadata poses a more fundamental challenge. When the groupings implied by metadata do not respect the partial order of the true type DAG (i.e., the induced coarsening does not yield a valid quotient DAG), type-consistency constraints may actively misdirect edge orientations. Developing robustness to such misspecification remains an important direction for future work.

A natural next step is to apply this framework to domains that exhibit strong redundancy and rich side information, such as gene regulatory network inference, where metadata may include DNA sequences or gene annotations. Future work could also explore scalability to larger graphs and extensions for active learning settings. Overall, this work illustrates how metadata-driven typing offers a principled path toward more identifiable and reusable causal discovery.

Author Contributions

PB conceptualized the method, developed the theory, implemented the codebase, ran the experiments, and wrote the paper. AD and DS supervised the project and provided feedback on the manuscript.

Acknowledgements

PB acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). DS acknowledges support from NSERC Discovery Grant RGPIN-2023-04869, and a Canada-CIFAR AI Chair. This research was enabled in part by computing resources and support provided by Mila – Quebec AI Institute.

References

Ahmed Abdulaal, Adamos Hadjivasiliou, Nina Montana-Brown, Tiantian He, Ayodeji Ijishakin, Ivana Drobnjak, Daniel Castro, and Daniel Alexander. Causal modelling agents: Causal graph discovery through synergising metadata-and data-driven reasoning. In *International Conference on Learning Representations*, volume 2024, pages 57559–57610, 2024.

Raj Agrawal, Chandler Squires, Karren Yang, Karthikeyan Shanmugam, and Caroline Uhler. Abcd-strategy: Budgeted experimental design for targeted causal structure discovery. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3400–3409. PMLR, 2019.

Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74 (1):47, 2002.

Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *nature*, 406(6794):378–382, 2000.

Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, and Huanhuan Chen. Causal structure learning supervised by large language model. *arXiv preprint arXiv:2311.11689*, 2023.

Tobias Boege, Kaie Kubjas, Pratik Misra, and Liam Solus. Colored gaussian dag models. *arXiv preprint arXiv:2404.04024*, 2024.

Philippe Brouillard, Perouz Taslakian, Alexandre Lacoste, Sébastien Lachapelle, and Alexandre Drouin. Typing assumptions improve identification in causal discovery. In *Conference on Causal Learning and Reasoning*, pages 162–177. PMLR, 2022.

Florian Peter Busch, Moritz Willig, Florian Guldan, Kristian Kersting, and Devendra Singh Dhami. Tagged for direction: Pinning down causal edge directions with precision. *arXiv preprint arXiv:2506.19459*, 2025.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Kristy Choi, Chris Cundy, Sanjari Srivastava, and Stefano Ermon. Lmpriors: Pre-trained language models as task-specific priors. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022.

Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl_1):D258–D261, 2004.

Shuyu Dong and Michèle Sebag. From graphs to dags: a low-complexity model and a scalable algorithm. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 107–122. Springer, 2022.

Xinshuai Dong, Biwei Huang, Ignavier Ng, Xiangchen Song, Yujia Zheng, Songyao Jin, Roberto Legaspi, Peter Spirtes, and Kun Zhang. A versatile causal discovery framework to allow causally-related hidden variables. In *International Conference on Learning Representations*, volume 2024, pages 43084–43118, 2024.

Dorit Dor and Michael Tarsi. A simple algorithm to construct a consistent extension of a partially oriented graph. *Technicial Report R-185, Cognitive Systems Laboratory, UCLA*, page 45, 1992.

Gerald M Edelman and Joseph A Gally. Degeneracy and complexity in biological systems. *Proceedings of the national academy of sciences*, 98(24):13763–13768, 2001.

Paul Erdos, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.

Zhuangyan Fang, Shengyu Zhu, Jiji Zhang, Yue Liu, Zhitang Chen, and Yangbo He. On low-rank directed acyclic graphs and causal structure learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Nir Friedman and Daphne Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, 50:95–125, 2003.

Nir Friedman, Moises Goldszmidt, and Abraham Wyner. Data analysis with bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 196–205, 1999.

Mariel Goddu and Caren M Walker. Certain to be surprised: A preference for novel causal outcomes develops in early childhood. In *CogSci*, 2020.

Alexander Hägele, Jonas Rothfuss, Lars Lorch, Vignesh Ram Somnath, Bernhard Schölkopf, and Andreas Krause. Bacadi: Bayesian causal discovery with unknown interventions. In *International Conference on Artificial Intelligence and Statistics*, pages 1411–1436. PMLR, 2023.

Søren Højsgaard and Steffen L Lauritzen. Graphical gaussian models with edge and vertex symmetries. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(5):1005–1027, 2008.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, volume 3, page 5, 2006.

Charles Kemp, Noah D Goodman, and Joshua B Tenenbaum. Learning to learn causal models. *Cognitive Science*, 34 (7):1185–1243, 2010.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*, 2024.

Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Identifiability of deep generative models without auxiliary information. *Advances in Neural Information Processing Systems*, 35:15687–15701, 2022.

Elizabeth Lapidow, Amberley R Stein, and Caren M Walker. Children use causality to guide question asking. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45, 2023.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. Causal discovery with language models as imperfect experts. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling (SPIGM)*, 2023.

Romain Lopez, Jan-Christian Hütter, Jonathan Pritchard, and Aviv Regev. Large-scale differentiable causal discovery of factor graphs. *Advances in Neural Information Processing Systems*, 35:19290–19303, 2022.

Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. Dibs: Differentiable bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

Marc Maier, Katerina Marazopoulou, David Arbour, and David Jensen. A sound and complete algorithm for learning causal models from relational data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 371–380, 2013.

Visu Makam, Philipp Reichenbach, and Anna Seigal. Symmetries in directed gaussian graphical models. *Electronic Journal of Statistics*, 17(2):3969–4010, 2023.

Vikash Mansinghka, Charles Kemp, Thomas Griffiths, and Joshua Tenenbaum. Structured priors for structure learning. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 324–331, 2006.

Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6(1):355–378, 2019.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 403–410, 1995.

OpenAI. Gpt-4 technical report. Technical report, OpenAI, 2023. URL https://cdn.openai.com/papers/gpt-4.pdf.

Jonas Peters and Peter Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT press, 2017.

Hoifung Poon, Chris Quirk, Charlie DeZiel, and David Heckerman. Literome: Pubmed-scale genomic knowledge base in the cloud. *Bioinformatics*, 30(19):2840–2842, 2014.

Parjanya Prashant, Ignavier Ng, Kun Zhang, and Biwei Huang. Differentiable causal discovery for latent hierarchical causal models. In *International Conference on*

Learning Representations, volume 2025, pages 31850–31875, 2025.

Alexandra Rett, Jamie Amemiya, Micah Goldwater, and Caren M Walker. Children's use of causal structure when making similarity judgments. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43, 2021.

Alexandra Rett, Micah Goldwater, and Caren M Walker. Children's recognition of shared causal structure in mechanical systems. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44, 2022.

Babak Salimi, Harsh Parikh, Moe Kayali, Lise Getoor, Sudeepa Roy, and Dan Suciu. Causal relational learning. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data*, pages 241–256, 2020.

Alexander F Siegenfeld and Yaneer Bar-Yam. An introduction to complex systems science and its applications. *Complexity*, 2020(1):6105872, 2020.

Liam Solus, Yuhao Wang, and Caroline Uhler. Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika*, 108(4):795–814, 2021.

Yong-Ling Song and Su-Shing Chen. Text mining biomedical literature for constructing gene regulatory networks. *Interdisciplinary Sciences: Computational Life Sciences*, 1:179–186, 2009.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Olaf Sporns and Richard F Betzel. Modular brain networks. *Annual review of psychology*, 67:613–640, 2016.

Ruibo Tu, Kun Zhang, Bo Bertilson, Hedvig Kjellstrom, and Cheng Zhang. Neuropathic pain diagnosis simulator for causal discovery algorithm evaluation. *Advances in Neural Information Processing Systems*, 32, 2019.

Ruibo Tu, Chao Ma, and Cheng Zhang. Causal-discovery performance of chatgpt in the context of neuropathic pain diagnosis. *arXiv preprint arXiv:2301.13819*, 2023.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008.

Tom S Verma and Judea Pearl. On the equivalence of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 220–227, 1990.

Andreas Wagner. Distributed robustness versus redundancy as causes of mutational robustness. *Bioessays*, 27(2): 176–188, 2005.

Caren M Walker and Alison Gopnik. Toddlers infer higher-order relational principles in causal learning. *Psychological science*, 25(1):161–169, 2014.

James M Whitacre. Degeneracy: a link between evolvability, robustness and complexity in biological systems. *Theoretical Biology and Medical Modelling*, 7(1):6, 2010.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Jun Wu and Mathias Drton. Partial homoscedasticity in causal discovery with linear models. *IEEE Journal on Selected Areas in Information Theory*, 2023.

Cheng Zhang, Hedvig Kjellström, Carl Henrik Ek, and Bo Bertilson. Diagnostic prediction using discomfort drawings with ibtm. In *Machine Learning for Healthcare Conference*, pages 226–238. PMLR, 2016.

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 804–813, 2011.

Rebecca Zhu and Alison Gopnik. Preschoolers and adults learn from novel metaphors. *Psychological Science*, 34 (6):696–704, 2023.

SUPPLEMENTARY MATERIAL

A DERIVATIONS OF THE MAIN RESULTS

A.1 DERIVATION OF THE LATENT POSTERIOR

In this section, we show the derivation of the posterior expectation with the continuous latent from Prop 1:

Proposition 1. *Following the proposed generative model, the expectation* $\mathbb{E}_{p(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}[f(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})]$ *with discrete variable is equal to:*

$$\mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}\left[\frac{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[wf(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})]}{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[w]}\right] \tag{15}$$

where $w := p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Psi} \mid \boldsymbol{T})$.

We obtain an equivalent expectation written as a normalized inner expectation.

$$\mathbb{E}_{p(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi}|\boldsymbol{M},\boldsymbol{X})}[f(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})] =$$

$$= \sum_{\boldsymbol{S},\boldsymbol{T}} \int_{\boldsymbol{\Theta},\boldsymbol{\Psi}} p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T} \mid \boldsymbol{M},\boldsymbol{X}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})\, d\boldsymbol{\Theta}\, d\boldsymbol{\Psi}$$

$$= \sum_{\boldsymbol{S},\boldsymbol{T}} \int_{\boldsymbol{\Theta},\boldsymbol{\Psi}} \frac{p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T},\boldsymbol{M},\boldsymbol{X}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})}{p(\boldsymbol{M},\boldsymbol{X})}\, d\boldsymbol{\Theta}\, d\boldsymbol{\Psi}$$

$$= \sum_{\boldsymbol{S},\boldsymbol{T}} \int_{\boldsymbol{\Theta},\boldsymbol{\Psi}} \int_{\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}}} \frac{p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T},\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{M},\boldsymbol{X}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})}{p(\boldsymbol{M},\boldsymbol{X})}\, d\boldsymbol{\Theta}\, d\boldsymbol{\Psi}\, d\tilde{\boldsymbol{S}}\, d\tilde{\boldsymbol{T}}$$

(we introduced the latent $\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}}$)

$$= \sum_{\boldsymbol{S},\boldsymbol{T}} \int_{\boldsymbol{\Theta},\boldsymbol{\Psi}} \int_{\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}}} \frac{p(\tilde{\boldsymbol{S}})p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}})p(\tilde{\boldsymbol{T}})p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})}{p(\boldsymbol{M},\boldsymbol{X})}\, d\boldsymbol{\Theta}\, d\boldsymbol{\Psi}\, d\tilde{\boldsymbol{S}}\, d\tilde{\boldsymbol{T}}$$

(we expanded the joint distribution)

$$= \sum_{\boldsymbol{S},\boldsymbol{T}} \int_{\boldsymbol{\Theta},\boldsymbol{\Psi}} \int_{\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}}} \frac{p(\boldsymbol{\Theta},\boldsymbol{\Psi},\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}} \mid \boldsymbol{M},\boldsymbol{X})p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}})p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})}{p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X} \mid \tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}})}$$

(we used the fact that $p(\boldsymbol{M},\boldsymbol{X}) = p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X} \mid \tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}})/p(\boldsymbol{\Theta},\boldsymbol{\Psi},\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}} \mid \boldsymbol{M},\boldsymbol{X})p(\tilde{\boldsymbol{S}})p(\tilde{\boldsymbol{T}})$)

$$= \mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}\left[\frac{\sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}})p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})}{p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X} \mid \tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}})}\right]$$

$$= \mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}\left[\frac{\sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}})p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})}{\sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X},\boldsymbol{S},\boldsymbol{T} \mid \tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}})}\right]$$

(by the law of total probability)

$$= \mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}\left[\frac{\sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}})p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})}{\sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}})p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T})}\right]$$

(expansion of the denominator following our generative model)

$$= \mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}\left[\frac{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})} p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T}) f(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{S},\boldsymbol{T})}{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})} p(\boldsymbol{\Psi} \mid \boldsymbol{T})p(\boldsymbol{M} \mid \boldsymbol{T},\boldsymbol{\Psi})p(\boldsymbol{X} \mid \boldsymbol{\Theta},\boldsymbol{S},\boldsymbol{T})p(\boldsymbol{\Theta} \mid \boldsymbol{S},\boldsymbol{T})}\right]$$

A.2 DERIVATION OF SCORES

Here are the derivation of the scores such as the one presented in Prop 2. Since they are highly similar, we only included comments for the first derivation.

A.2.1 Score of $\tilde{S}$ and $\tilde{T}$

Proposition 2. *The score of the log-posterior* $\nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \mathcal{D})$ *is equal to:*

$$\nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}) + \frac{\nabla_{\tilde{\boldsymbol{S}}} \mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[w]}{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[w]} \tag{16}$$

with w *as defined in Proposition 1.*

Let $S_{\tilde{\boldsymbol{S}}} := \nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \boldsymbol{M}, \boldsymbol{X})$.

$$S_{\tilde{\boldsymbol{S}}} = \nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X}) - \nabla_{\tilde{\boldsymbol{S}}} \log p(\boldsymbol{M}, \boldsymbol{X}) \tag{18}$$

The last term is removed since it does not depend on $\tilde{\boldsymbol{S}}$:

$$S_{\tilde{\boldsymbol{S}}} = \nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X}) \tag{19}$$

$$= \nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}) + \nabla_{\tilde{\boldsymbol{S}}} \log p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}) \tag{20}$$

Using the fact that $\nabla_x \log f(x) = \frac{\nabla_x f(x)}{f(x)}$:

$$S_{\tilde{\boldsymbol{S}}} = \nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}) + \frac{\nabla_{\tilde{\boldsymbol{S}}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}})}{p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}})} \tag{21}$$

$$= \nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}) + \frac{\nabla_{\tilde{\boldsymbol{S}}} \sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T}) p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}}) p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})}{\sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T}) p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}}) p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})} \tag{22}$$

$$= \nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}) + \frac{\nabla_{\tilde{\boldsymbol{S}}} \mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T})]}{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T})]}. \tag{23}$$

We show in the next section how this score can be approximated using samples using either reinforce or the Gumbel-softmax trick. Similarly, for the score of the types, let $S_{\tilde{\boldsymbol{T}}} := \nabla_{\tilde{\boldsymbol{T}}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \boldsymbol{M}, \boldsymbol{X})$. By using the same derivation technique, we get:

$$S_{\tilde{\boldsymbol{T}}} = \nabla_{\tilde{\boldsymbol{T}}} \log p(\tilde{\boldsymbol{T}}) + \frac{\nabla_{\tilde{\boldsymbol{T}}} \mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T})]}{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T})]}. \tag{24}$$

A.2.2 Score of parameters

For the score of parameters $\boldsymbol{\Theta}$, let $S_{\boldsymbol{\Theta}} := \nabla_{\boldsymbol{\Theta}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \boldsymbol{M}, \boldsymbol{X})$.

$$S_{\boldsymbol{\Theta}} = \nabla_{\boldsymbol{\Theta}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X}) - \nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{M}, \boldsymbol{X}) \tag{25}$$

$$= \nabla_{\boldsymbol{\Theta}} \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X}) \tag{26}$$

$$= \nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}) \tag{27}$$

$$= \frac{\nabla_{\boldsymbol{\Theta}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}})}{p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}})} \tag{28}$$

$$= \frac{\nabla_{\boldsymbol{\Theta}} \sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T}) p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}}) p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})}{\sum_{\boldsymbol{S},\boldsymbol{T}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T}) p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}}) p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})} \tag{29}$$

$$= \frac{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[\nabla_{\boldsymbol{\Theta}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T})]}{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}), p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T})]}. \tag{30}$$

The score of $\boldsymbol{\Psi}$ follows the same development.

A.2.3 Score of priors

The score of the latent prior $\nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}})$ is:

$$\nabla_{\tilde{\boldsymbol{S}}} \log p(\tilde{\boldsymbol{S}}) = \nabla_{\tilde{\boldsymbol{S}}} \log \left(\prod_{ij} \mathcal{N}(\tilde{S}_{ij}; 0, \sigma_{\tilde{S}}^2) \right) = -\frac{1}{\sigma_{\tilde{S}}^2} \tilde{\boldsymbol{S}}. \tag{31}$$

The score of the latent prior $\nabla_{\tilde{\boldsymbol{T}}} \log p(\tilde{\boldsymbol{T}})$ is:

$$\nabla_{\tilde{\boldsymbol{T}}} \log p(\tilde{\boldsymbol{T}}) = \sum_{i=1}^{d} \sum_{j=1}^{k} (\xi_j - 1) \nabla_{\tilde{\boldsymbol{T}}} \log \eta_j - \frac{1}{\sigma_{\tilde{T}}^2} \tilde{\boldsymbol{T}}, \tag{32}$$

where $\eta_j = \text{softmax}_\gamma(\tilde{\boldsymbol{T}}_j)$. In practice, we use a similar trick to Hägele et al. [2023] where we clip the values of $\boldsymbol{\eta}$ that are outside the range $(1e^{-8}, 1e^{8})$.

A.3 ANNEALING

Proposition 3. *Let α and γ denote the inverse temperatures of the Bernoulli and categorical relaxations for the skeleton and types, respectively. As $\alpha, \gamma \to \infty$, we have that*

$$\mathbb{E}_{p(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}[f(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})] \;\to\; \mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}[f(\boldsymbol{S}_\infty(\tilde{\boldsymbol{S}}),\boldsymbol{T}_\infty(\tilde{\boldsymbol{T}}),\boldsymbol{\Theta},\boldsymbol{\Psi})]$$

where $\boldsymbol{S}_\infty(\tilde{\boldsymbol{S}})_{ij} := \mathbf{1}_{\tilde{\boldsymbol{S}}_{ij}>0}$ and $\boldsymbol{T}_\infty(\tilde{\boldsymbol{T}})_i := \text{onehot}\Big(\arg\max_j \tilde{\boldsymbol{T}}_{ij}\Big)$.

Proof. The proof follows the annealing arguments of Lorch et al. [2021] and Hägele et al. [2023]. First, recall that from Proposition 1, we have

$$\mathbb{E}_{p(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}[f(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})] = \mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})} \left[\frac{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})} w f(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})}{\mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})} w} \right].$$

As $\alpha, \gamma \to \infty$, the sigmoid and softmax relaxations converge to hard thresholding and one-hot assignments, respectively, such that $p(\boldsymbol{S} \mid \tilde{\boldsymbol{S}})$ and $p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}})$ concentrate on deterministic limits $\boldsymbol{S}_\infty(\tilde{\boldsymbol{S}})$ and $\boldsymbol{T}_\infty(\tilde{\boldsymbol{T}})$. Consequently, for any bounded function f, we have

$$\mathbb{E}_{p(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}[f(\boldsymbol{S},\boldsymbol{T},\boldsymbol{\Theta},\boldsymbol{\Psi})] \;\to\; \mathbb{E}_{p(\tilde{\boldsymbol{S}},\tilde{\boldsymbol{T}},\boldsymbol{\Theta},\boldsymbol{\Psi}|\mathcal{D})}[f(\boldsymbol{S}_\infty(\tilde{\boldsymbol{S}}),\boldsymbol{T}_\infty(\tilde{\boldsymbol{T}}),\boldsymbol{\Theta},\boldsymbol{\Psi})] \tag{33}$$

since the inner expectation of Proposition 1 converges to a single point and cancels out.

□

A.4 MONTE CARLO ESTIMATES

If the score function is used to approximate the numerator of Eq. 16, then we get:

$$\nabla_{\tilde{\boldsymbol{T}}} \mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X} \mid \boldsymbol{S},\boldsymbol{T})] =$$

$$= \mathbb{E}_{\boldsymbol{S}} \sum_{\boldsymbol{T}} \nabla_{\tilde{\boldsymbol{T}}} p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}}) p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X} \mid \boldsymbol{S},\boldsymbol{T}) \tag{34}$$

$$= \mathbb{E}_{\boldsymbol{S}} \sum_{\boldsymbol{T}} p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}}) \nabla_{\tilde{\boldsymbol{T}}} \log p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}}) p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X} \mid \boldsymbol{S},\boldsymbol{T}) \tag{35}$$

$$= \mathbb{E}_{\boldsymbol{S},\boldsymbol{T}} \Big[p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X} \mid \boldsymbol{S},\boldsymbol{T}) \nabla_{\tilde{\boldsymbol{T}}} \log p(\boldsymbol{T} \mid \tilde{\boldsymbol{T}}) \Big] \tag{36}$$

$$\approx \sum_{i,j}^{b} p(\boldsymbol{\Theta},\boldsymbol{\Psi},\boldsymbol{M},\boldsymbol{X} \mid \boldsymbol{S}^{(i)},\boldsymbol{T}^{(j)}) \nabla_{\tilde{\boldsymbol{T}}} \log p(\boldsymbol{T}^{(j)} \mid \tilde{\boldsymbol{T}}), \tag{37}$$

where b is the number of examples in a batch used to do the Monte Carlo estimation. To ensure numerical stability, in practice we estimate this using the logsumexp trick.

If the Gumbel-softmax trick [Jang et al., 2017, Maddison et al., 2017] is used to approximate the numerator, then we get:

$$\nabla_{\tilde{\boldsymbol{S}}} \mathbb{E}_{p(\boldsymbol{S}|\tilde{\boldsymbol{S}}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T})]$$

$$\approx \mathbb{E}_{\boldsymbol{L}\sim p(\boldsymbol{L}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[\nabla_{\tilde{\boldsymbol{G}}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid f(\boldsymbol{L}, \tilde{\boldsymbol{S}}), \boldsymbol{T})] \tag{38}$$

$$= \mathbb{E}_{\boldsymbol{L}\sim p(\boldsymbol{L}),p(\boldsymbol{T}|\tilde{\boldsymbol{T}})}[\nabla_{\boldsymbol{S}} p(\boldsymbol{\Theta}, \boldsymbol{\Psi}, \boldsymbol{M}, \boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{T})|_{\boldsymbol{S}=f(\boldsymbol{L},\tilde{\boldsymbol{S}})} \nabla_{\tilde{\boldsymbol{S}}} f(\boldsymbol{L}, \tilde{\boldsymbol{S}})] \tag{39}$$

where $f(\boldsymbol{L}, \tilde{\boldsymbol{S}}) := \sigma(\boldsymbol{L} + \tilde{\boldsymbol{S}})$ and $L \sim \text{Logistic}(0, 1)$.

B STEIN VARIATIONAL GRADIENT DESCENT METHOD

B.1 BACKGROUND

The Stein Variational Gradient Descent (SVGD) [Liu and Wang, 2016] is a general algorithm to perform Bayesian inference. At a high level, the algorithm starts from a distribution q that it iteratively transforms to match a target distribution p. It uses a fixed set of particles that it iteratively moves following gradient descent updates. Relying on kernels k, it also repulses particles that are too close together, making sure that the different particles cover the target distribution and do not suffer from a mode collapse. Before presenting the actual algorithm, we present its theoretical foundation.

Stein's identity. Let $p(x)$ be a smooth (continuously differentiable) density and $\boldsymbol{\phi}(x)$ a smooth vector function. Then, the Stein's identity is:

$$\mathbb{E}_{x\sim p}[\mathcal{A}_p \boldsymbol{\phi}(x)] = 0 \tag{40}$$

where $\mathcal{A}_p$ is the Stein operator that returns a zero mean function when $x \sim p$:

$$\mathcal{A}_p := \nabla_x \log p(x) \boldsymbol{\phi}(x)^T + \nabla_x \boldsymbol{\phi}(x). \tag{41}$$

Stein's discrepancy. From Stein's identity, one can define a discrepancy between two distributions p and q using $\mathbb{E}_{x\sim q}[\mathcal{A}_p \boldsymbol{\phi}(x)]$. For different p and q, this will be different from 0. The Stein discrepancy is the maximum violation of the Stein's identity:

$$\mathbb{D}(q, p) := \max_{\boldsymbol{\phi}\in\mathcal{F}} \{\mathbb{E}_{x\sim q}[\text{trace}(\mathcal{A}_p \boldsymbol{\phi}(x))]\} \tag{42}$$

Kernelized Stein's discrepancy. The set of functions $\mathcal{F}$ will greatly impact the computational tractability of the Stein discrepancy. The Kernelized Stein discrepancy lead to closed form solution by maximizing $\boldsymbol{\phi}$ in the unit ball of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$.

$$\mathbb{D}(q, p) = \max_{\boldsymbol{\phi}\in\mathcal{H}} \{\mathbb{E}_{x\sim q}[\text{trace}(\mathcal{A}_p \boldsymbol{\phi}(x)) \text{ st } ||\boldsymbol{\phi}||_{\mathcal{H}\leq 1}\}. \tag{43}$$

The optimal solution of Eq. 43 is, up to renormalization, given by:

$$\boldsymbol{\phi}^*_{q,p}(\cdot) = \mathbb{E}_{x\sim q}[\mathcal{A}_p k(x, \cdot)] \tag{44}$$

where k is a kernel in the RKHS $\mathcal{H}$.

Link with KL. Let $\mathcal{Q}$ be the family of distributions defined as the smooth transform of q_0 defined as:

$$T(x) = x + \eta \boldsymbol{\phi}^*(x). \tag{45}$$

Then, Liu and Wang [2016] show that $\boldsymbol{\phi}^*$ is the direction of the steepest descent of the KL divergence between p and q.

SVGD algorithm. Based on their theoretical result, Liu and Wang [2016] have proposed the SVGD algorithm which consists of first sampling particles $\{\boldsymbol{x}^{(k)}\}_{l=1}^L$, according to some prior, and then iteratively applying this transformation:

$$\boldsymbol{x}^{(l)}_{t+1} \leftarrow \boldsymbol{x}^{(l)}_t + \eta \boldsymbol{\phi}(\boldsymbol{x}^{(l)}_t) \tag{46}$$

where the function $\boldsymbol{\phi}$ is:

$$\boldsymbol{\phi}(\cdot) = \frac{1}{L}\sum_{l=1}^{L}\left[k(\boldsymbol{x}_t^{(l)}, \cdot)\nabla_{\boldsymbol{x}_t^{(l)}} \log p(\boldsymbol{x}_t^{(l)}) + \nabla_{\boldsymbol{x}_t^{(l)}} k(\boldsymbol{x}_t^{(l)}, \cdot)\right]. \tag{47}$$

The score $\nabla_{\boldsymbol{x}_t^{(l)}} \log p(\boldsymbol{x}_t^{(l)})$ drives the particle towards high-density regions of p while the term $\nabla_{\boldsymbol{x}_t^{(l)}} k(\boldsymbol{x}_t^{(l)}, \cdot)$ is a repulsive force pushing the particles away from each others. If η is sufficiently small, the particles will converge, i.e., the transform in Eq. 46 will be the identity.

The key discovery behind the SVGD algorithm is that the transform $\boldsymbol{\phi}$ in Eq. 47 is optimal to reduce the KL-divergence between p and q.

B.2 ALGORITHM

In this section, we show in detail the SVGD algorithm used for our application. To simplify notation, we use $\{\boldsymbol{\Omega}_0^{(l)}\}_{l=1}^L := \{(\tilde{\boldsymbol{S}}^{(l)}, \tilde{\boldsymbol{T}}^{(l)}, \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Psi}^{(l)})\}_{l=1}^L$ as a way to represent the different particles succinctly.

Algorithm 1 t-DiBS with SVGD for inference of $p(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \mathcal{D})$

Input: Dataset $\mathcal{D}$, $\boldsymbol{M}$, Kernel k, schedules for $\alpha_t, \beta_t, \gamma_t$ and stepsize η
Output: Set of particles $\{(\boldsymbol{S}^{(l)}, \boldsymbol{T}^{(l)}, \boldsymbol{\Theta}^{(l)}, \boldsymbol{\Psi}^{(l)})\}_{l=1}^L$

Initialize a set of latent and parameter particles $\{\boldsymbol{\Omega}_0^{(l)}\}_{l=1}^L$
for $t = 0$ **to** $T-1$ **do**
  Estimate the score $\nabla \log p(\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi} \mid \mathcal{D})$ for each particle
  for particle $l = 1$ **to** L **do**
    $\tilde{\boldsymbol{S}}_{t+1}^{(l)} \leftarrow \tilde{\boldsymbol{S}}_t^{(l)} + \eta_t \boldsymbol{\phi}_t^{\tilde{\boldsymbol{S}}}(\boldsymbol{\Omega}_t^{(l)})$
    $\tilde{\boldsymbol{T}}_{t+1}^{(l)} \leftarrow \tilde{\boldsymbol{T}}_t^{(l)} + \eta_t \boldsymbol{\phi}_t^{\tilde{\boldsymbol{T}}}(\boldsymbol{\Omega}_t^{(l)})$
    $\boldsymbol{\Theta}_{t+1}^{(l)} \leftarrow \boldsymbol{\Theta}_t^{(l)} + \eta_t \boldsymbol{\phi}_t^{\boldsymbol{\Theta}}(\boldsymbol{\Omega}_t^{(l)})$
    $\boldsymbol{\Psi}_{t+1}^{(l)} \leftarrow \boldsymbol{\Psi}_t^{(l)} + \eta_t \boldsymbol{\phi}_t^{\boldsymbol{\Psi}}(\boldsymbol{\Omega}_t^{(l)})$
    where $\boldsymbol{\phi}_t^{\boldsymbol{\Theta}}(\cdot) := \frac{1}{M}\sum_{m=1}^{M}\left[k(\boldsymbol{\Omega}_t^{(m)}, \cdot)\nabla_{\boldsymbol{\Theta}_t^{(m)}} \log p(\boldsymbol{\Omega}_t^{(m)} \mid \mathcal{D}) + \nabla_{\boldsymbol{\Theta}_t^{(m)}} k(\boldsymbol{\Omega}_t^{(m)}, \cdot)\right]$
    and $\boldsymbol{\phi}_t^{\tilde{\boldsymbol{S}}}, \boldsymbol{\phi}_t^{\tilde{\boldsymbol{T}}}, \boldsymbol{\phi}_t^{\boldsymbol{\Psi}}$ are similar to $\boldsymbol{\phi}_t^{\boldsymbol{\Theta}}$ but use $\nabla_{\tilde{\boldsymbol{S}}_t^{(m)}}, \nabla_{\tilde{\boldsymbol{T}}_t^{(m)}}, \nabla_{\boldsymbol{\Psi}_t^{(m)}}$
  end for
end for
return $\{(\boldsymbol{S}_\infty(\tilde{\boldsymbol{S}}_T^{(l)}), \boldsymbol{T}_\infty(\tilde{\boldsymbol{T}}_T^{(l)}), \boldsymbol{\Theta}_T^{(l)}, \boldsymbol{\Psi}_T^{(l)})\}_{l=1}^L$

In our experiments, we use the additive RBF kernel for k presented in the next section. The specific hyperparameters used are described in Appendix F.2.

B.3 KERNEL

As Lorch et al. [2021] we use an additive RBF kernel for our SVGD implementation:

$$\begin{aligned} k((\tilde{\boldsymbol{S}}, \tilde{\boldsymbol{T}}, \boldsymbol{\Theta}, \boldsymbol{\Psi}), (\tilde{\boldsymbol{S}}', \tilde{\boldsymbol{T}}', \boldsymbol{\Theta}', \boldsymbol{\Psi}')) := h_{\tilde{S}} \exp\left(-\frac{||\tilde{\boldsymbol{S}} - \tilde{\boldsymbol{S}}'||^2}{2\sigma_{\tilde{S}}}\right) + h_{\tilde{T}} \exp\left(-\frac{||\tilde{\boldsymbol{T}} - \tilde{\boldsymbol{T}}'||^2}{2\sigma_{\tilde{T}}}\right) \\ + h_{\Theta} \exp\left(-\frac{||\boldsymbol{\Theta} - \boldsymbol{\Theta}'||^2}{2\sigma_{\Theta}}\right) + h_{\Psi} \exp\left(-\frac{||\boldsymbol{\Psi} - \boldsymbol{\Psi}'||^2}{2\sigma_{\Psi}}\right), \end{aligned}$$

where the $h > 0$ are scaling of each kernel and σ their bandwidth.

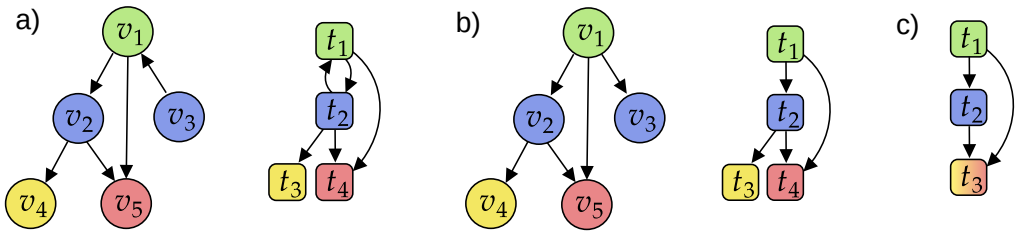


Figure 6: In a), a representation of an inconsistent t-DAG (left) and its corresponding type graph (right). Here, colors represent types. Similarly, in b), a consistent t-DAG with its type DAG. In c), the minimal type DAG of the t-DAG in b.

C THEORETICAL RESULTS ABOUT GRAPHS

C.1 DEFINITIONS

In this section, we provide additional definitions about t-DAGs and f-DAGs that will be useful for the theoretical results.

Definition 3 (Type graph)**.** *For a given t-DAG G_d^k, T, a type graph is a directed graph $G = (V, E)$ with $V = \{t_1, \dots, t_k\}$ and E is the set of all t-edges of G_d^k, T (where $t_i \to t_j$ is a t-edge if there exists $u \to v$ in G_d^k with $T(u) = t_i$ and $T(v) = t_j$).*

Definition 4 (Type DAG)**.** *A type DAG is a type graph that is acyclic.*

Definition 5 (Minimal type DAG)**.** *Given a t-DAG G_d^k, a minimal type DAG is defined as the type DAG with the minimal number of nodes ($|\mathcal{T}'| \leq k$) where $\mathcal{T}'$ is a partitioning of $\mathcal{T}$ such that the t-edges between the types $\mathcal{T}'$ are still type consistent.*

We also define the notion of semantic layer in Def 6. Intuitively, semantic layers capture successive stages of causal influence: variables in higher layers can affect only variables in lower layers, while variables within the same layer do not causally interact.

Definition 6 (Semantic layers)**.** *Let $G_{\mathcal{T}}$ denote the type DAG of a t-DAG. The semantic layer of a type t is its height in $G_{\mathcal{T}}$, defined as the length of the longest directed path ending at t. The semantic layer of a variable v is the layer of its type $\mathcal{T}(v)$.*

We show in Fig. 6 visual examples of the definition related to t-DAG. In Fig. 6a, we show an example of a DAG that is not type consistent with its corresponding type graph. It is inconsistent since, using $T(v_1) = t_i$ and $T(v_2) = T(v_3) = t_j$, both $E(t_i, t_j) \neq \emptyset \neq E(t_j, t_i)$. We show in Fig. 6b a consistent t-DAG with its corresponding type DAG. In Fig. 6c, we show the minimal type DAG for the t-DAG in b. In that case, the yellow and red types are binned together since they are in the same tier.

We introduce the notion of height both for a specific vertex and for a DAG.

Definition 7 (Height of a vertex)**.** *The height of a vertex, $h(v_i) \in \mathbb{N}$, is defined as the length of the longest directed path starting at v_i.*

Definition 8 (Height of a DAG)**.** *The height of a DAG $G = (V, E)$ corresponds to the length of its longest path, i.e., $\max_i h(v_i)$.*

We also introduce the notion of quotient graph and quotient DAG. We write $[v] = \{u \in V : u \sim v\}$ for the equivalence class of v under $\sim$.

Definition 9 (Quotient graph)**.** *Let $G = (V, E)$ be a directed graph and $\sim$ an equivalence relation on V. The* quotient graph *$G/\!\sim$ is the directed graph with vertex set $V/\!\sim = \{[v] : v \in V\}$ and edge set*

$$E/\!\sim \;=\; \{([u], [v]) : \exists\, u' \in [u],\, v' \in [v] \text{ s.t. } (u', v') \in E \text{ and } [u] \neq [v]\},$$

where $[v]$ denotes the equivalence class of v under $\sim$.

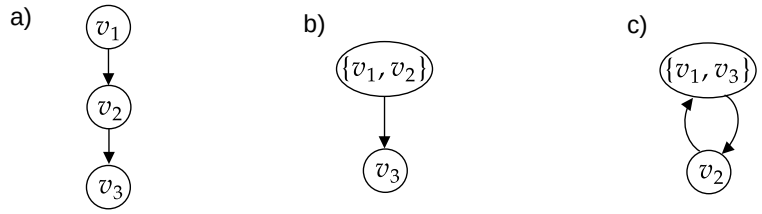


Figure 7: Examples of a DAG (a) and two quotient graphs obtained by collapsing different equivalence classes. In (b), merging v_1 and v_2 respects the partial order of the DAG, yielding a valid quotient DAG. In (c), merging v_1 and v_3 violates the partial order (v_2 lies on a directed path between them) inducing a cycle in the quotient, thus failing to produce a DAG.

Definition 10 (Quotient DAG)**.** *Let $G = (V, E)$ be a DAG and $\sim$ an equivalence relation on V. The quotient graph $G/\sim$ is a* quotient DAG *if it is acyclic, i.e., if $\sim$ respects the partial order induced by G. This holds if and only if there is no equivalence class $[v]$ such that a directed path exists from $[v]$ to $[v]$ in $G/\sim$, or equivalently, if no vertex outside an equivalence class lies on a directed path between two vertices within it.*

In Fig. 7, we illustrate the notion of quotient graph and DAG, showing examples of equivalence relations: one leading to a valid quotient DAG (Fig. 7.b) and another one that does not since its equivalence relation does not respect the partial order of G (Fig. 7.c).

We also note that the type graph of a t-DAG G with type assignment T is precisely the quotient graph $G/\sim$, where $\sim$ is the equivalence relation defined by $u \sim v \iff T(u) = T(v)$. Consequently, the type DAG is a quotient DAG in the sense of Definition 10, and the metadata coarsening introduced in Proposition 6 generalizes the type graph construction: rather than collapsing by the true type assignment, it collapses by the coarser partition induced by indistinguishable metadata distributions.

Finally, t-DAGs are closely related to factor DAGs (f-DAGs) introduced by Lopez et al. [2022], which represent causal structure using feature and factor nodes in a bipartite graph.

Definition 11 (f-DAG; from Lopez et al. [2022])**.** *Let $V = \{v_1, \dots, v_d\}$ denote the set of feature vertices, and $M = \{m_1, \dots, m_k\}$ the set of factor vertices. Let E be a set of directed edges, such that there are no edges between two nodes of same type. We define a Factor Directed Graph (f-DAG) as the bipartite graph $F = (V, M, E)$.*

In the following sections, when we refer to t-DAGs we will by default assume that they are consistent t-DAGs having a type DAG. We denote $\mathcal{G}_d$ the set of DAGs and $\mathcal{G}_d^k$ the set t-DAGs where d is the number of vertices and $k \leq d$ types. We also denote by $\mathcal{F}_d^k$ the set of f-DAGs with d vertices and k modules.

C.2 RESULTS

In the following propositions, we present relations between the sets of DAGs, t-DAGs and f-DAGs. As a rough summary, the set of f-DAGs is a subset of t-DAGs which is a subset of DAGs (i.e., $\mathcal{F}_d^{k-1} \subseteq \mathcal{G}_d^k \subseteq \mathcal{G}_d$). We show also that t-DAGs with k types are equivalent to DAGs of height $k-1$. Finally, we show that, as the number of vertices increases and the number of types is fixed, the space of DAGs increases greatly compared to the space of t-DAGs, and similarly the space of t-DAGs increases greatly compared to the space of f-DAGs.

Proposition 7 (t-DAG vs DAG space)**.** *$\mathcal{G}_d^k \subseteq \mathcal{G}_d$ where the equality is reached if and only if $k = d$.*

Proof. By the definition of the t-DAGs, we know that it is always a DAG, i.e., $\mathcal{G}_d^k \subseteq \mathcal{G}_d$.

If $k < d$, then $\mathcal{G}_d^k$ is necessarily a strict subset of $\mathcal{G}_d$. It is the case since, by the pigeonhole principle, at least two vertices have the same type and no edges can be added between them. Thus, the fully connected DAG is not included in $\mathcal{G}_d^k$.

If $k = d$, then $\mathcal{G}_d^k = \mathcal{G}_d$. It means that every $G \in \mathcal{G}_d$ is also $G \in \mathcal{G}_d^d$. It is clear that since each vertex has a different type, there is absolutely no constraint on the space of t-DAGs. □

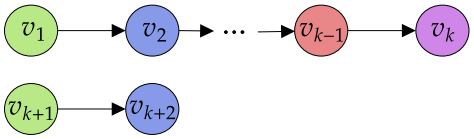


Figure 8: Example of a t-DAG with k types that cannot be represented as a f-DAG with k-1 modules. Here, the colors represent types.

Proposition 4 (Equivalence to bounded-height DAGs)**.** *Let $\mathcal{G}_d(\ell)$ and $\mathcal{G}_d^k$ be the sets of all DAGs of height ℓ and t-DAGs with d vertices and k types. Then, $\mathcal{G}_d^k = \bigcup_{\ell=0}^{k-1} \mathcal{G}_d(\ell)$.*

Proof. ($\mathcal{G}_d^k \subseteq \bigcup_{\ell=0}^{k-1} \mathcal{G}_d(\ell)$). We first show that every t-DAG is a DAG with a maximum length of $k-1$.

By definition, a t-DAG is a DAG. Furthermore, its longest possible path is $k-1$. To see this, by contradiction, let's assume that there is $G \in \mathcal{G}_d^k$ with a height of $\geq k$. A directed path of length k is constituted of $k+1$ vertices. This implies that at least two nodes along the path have the same type. Thus, two variables are both the ancestor and the descendant of this type, which is in contradiction with the fact that the type DAG is acyclic.

($\mathcal{G}_d^k \supseteq \bigcup_{\ell=0}^{k-1} \mathcal{G}_d(\ell)$). We now show that every DAG with a maximum height of $k-1$ have a t-DAG equivalent.

Since the maximum height of the DAG is $k-1$, each vertex has a height of $k-1$ or less (i.e., $h(v_i) \leq k-1, \forall i$). We can convert this DAG in a t-DAG with k types simply by assigning a different type for each possible height: $T(v_i) = t_{h(v_i)}$. □

Proposition 8 (t-DAG vs f-DAG)**.** *$\mathcal{F}_d^k \subseteq \mathcal{G}_d^{k+1}$ where the equality is only possible when $d \in \{k, k+1\}$.*

Proof. For every f-DAG $G \in \mathcal{F}_d^k$, we can show how to convert it into a t-DAG with at most $k+1$ types, thus showing that $\mathcal{F}_d^k \subseteq \mathcal{G}_d^{k+1}$. Let $\{v_1, \ldots, v_{i_1}\}, m_1, \{v_{i_1+1}, \ldots, v_{i_2}\}, m_2, \ldots, m_k, \{v_{i_k+1}, \ldots, v_d\}$ be an ordering of the vertices and the modules m_i. We denote by Π_i the vector of vertices directly preceding m_i. A vertex of Π_i can only be a parent to subsequent m_j ($j \geq i$), and it can only be a child to a previous m_j ($j < i$). Then, the vertices of each partition Π_i can be assigned to the type t_i. Here, since it is following a topological ordering, there is no possible cycle, and it is type consistent since members of a partition cannot cause each other, and they have a fixed directionality for every member of the other Π_j. Thus, the f-DAG can be represented as a t-DAG. An alternative way to show this result would be to show that the space of f-DAGs is a subset of DAGs with maximal height k and thus, a subset of t-DAGs with $k+1$ types.

For the other direction, we can generate examples for $d \geq k+2$ where a t-DAG G_d^{k+1} cannot be represented as a f-DAG F_d^k. We show a general example in Fig. 8. To generate this example, create a chain of length $k-1$ formed of $(v_1, \ldots, v_k)$. Then, for v_{k+1} and v_{k+2}, simply add an edge between them. First, we show that this graph is part of the t-DAGs set. Assume that the type DAG is the chain $(t_1, \ldots, t_k)$. Then we can assign the following types to each vertex and be type-consistent: $T(v_i) = t_i, \forall i \leq k$, and $T(v_{k+1}) = t_1, T(v_{k+2}) = t_2$. On the other hand, no f-DAG with $k-1$ modules exists. Assign a module for each vertex of the chain: $v_i \to m_i \to v_{i+1}, \forall i \leq k-1$. The last edge $v_{k+1} \to m_k \to v_{k+2}$ requires a new module m_k, but all the $k-1$ modules have already been assigned to the chain. □

Proposition 9 (Cardinality of DAGs spaces)**.** *For a fixed ordering σ, we have $|\mathcal{G}_d(\sigma)| = 2^{\frac{d(d-1)}{2}}$ and $|\mathcal{G}_d^k(\sigma)| \leq k^d \cdot 2^{\frac{(k-1)d^2}{2k}}$.*

Proof. Suppose that we have a fixed ordering σ such that w.l.o.g. $(v_1, \ldots, v_d)$. With this ordering, every vertex v_i can have outgoing edges to any v_j with $j > i$, that is $d-i$ possible edges. In total, there are $\sum_{i=1}^{d}(d-i) = \sum_{i=0}^{d-1} i = \frac{d(d-1)}{2}$ possible edges. Since each edge is independently present or absent, there are $2^{\frac{d(d-1)}{2}}$ possible DAGs with d vertices and ordering σ.

For the t-DAGs, we will first show that $|\mathcal{G}_d^k(\sigma)| \leq S(d,k) \cdot 2^{\sum_{l<j}^{k} |V_l||V_j|}$. We start by choosing a partitioning of the d vertices into k non-empty partitions $V_1, \ldots, V_k$, where partition V_l corresponds to the set of vertices assigned to type t_l. Since we assume that each type contains at least one variable, there are $S(d,k)$ such partitionings, where S denotes the

Stirling number of the second kind. By the type-consistency constraint, edges can only go from a partition of lower index to one of higher index. For a single vertex in partition V_l, there are $\sum_{j=l+1}^{k} |V_j|$ possible target vertices. Since each such edge is independently present or absent, the number of edge configurations originating from all vertices in V_l is $2^{|V_l| \sum_{j=l+1}^{k} |V_j|}$. Multiplying over all partitions, the total number of edge configurations for a given partitioning is $2^{\sum_{l<j}^{k} |V_l||V_j|}$.

In the following, we derive a looser but more interpretable upper bound. For the Stirling number, we use $S(d,k) \leq k^d$, since k^d counts all functions from d elements to k labels (including those that leave some labels unused), whereas $S(d,k)$ counts only surjections up to label ordering.

For the exponent, we show that:

$$\sum_{l<j}^{k} |V_l||V_j| \leq \sum_{l<j}^{k} \Big(\frac{d}{k}\Big)^2 = \frac{(k-1)d^2}{2k}. \tag{48}$$

That is, the sum $\sum_{l<j}^{k} |V_l||V_j|$ is maximized when all partitions have equal size. To see this, first note the identity:

$$\sum_{l<j}^{k} |V_l||V_j| = \frac{1}{2}\Big(\Big(\sum_{l=1}^{k} |V_l|\Big)^2 - \sum_{l=1}^{k} |V_l|^2\Big) = \frac{1}{2}\Big(d^2 - \sum_{l=1}^{k} |V_l|^2\Big), \tag{49}$$

where the last equality uses $\sum_{l=1}^{k} |V_l| = d$. This expression is maximized when $\sum_{l=1}^{k} |V_l|^2$ is minimized. By Jensen's inequality applied to the convex function $x \mapsto x^2$:

$$\frac{1}{k}\sum_{l=1}^{k} |V_l|^2 \geq \Big(\frac{1}{k}\sum_{l=1}^{k} |V_l|\Big)^2 = \frac{d^2}{k^2}. \tag{50}$$

Therefore:

$$\sum_{l<j}^{k} |V_l||V_j| \leq \frac{1}{2}\Big(d^2 - \frac{d^2}{k}\Big) = \frac{(k-1)d^2}{2k}. \tag{51}$$

Combining both terms yields the upper bound $k^d \cdot 2^{\frac{(k-1)d^2}{2k}}$. □

Proposition 5 (Limits of the cardinality, t-DAG vs DAG)**.** *The ratio* $\frac{|\mathcal{G}_d^k|}{|\mathcal{G}_d|}$ *converges to* 0 *when* $d \to \infty$ *and* k *is fixed.*

Proof. To calculate the limit of the ratio, we will use an upper bound of $|\mathcal{G}_d^k|$ and a lower bound $|\mathcal{G}_d|$. Even if these bounds are quite loose, they are sufficient to show that the ratio converges to 0.

We previously observed that, for a given ordering σ, $|\mathcal{G}_d^k(\sigma)| \leq \binom{d+k}{k} 2^{\frac{(k-1)d^2}{2k}}$. Thus, an upper bound of $|\mathcal{G}_d^k|$ is constructed simply by considering all the possible permutations of the different vertices, i.e., multiplying the previous bound by $d!$.

$$\begin{aligned}\frac{|\mathcal{G}_d^k|}{|\mathcal{G}_d|} &\leq d!\binom{d+k}{k} 2^{\frac{d^2(k-1)}{2k}} 2^{-\frac{d(d+1)}{2}} \\ &\leq \frac{(d+k)!}{k!} 2^{\frac{d^2(k-1)}{2k} - \frac{d(d+1)}{2}}.\end{aligned} \tag{52}$$

We use the non-asymptotic version of the Stirling formula for the factorial terms:

$$\begin{aligned}\frac{|\mathcal{G}_d^k|}{|\mathcal{G}_d|} &\leq \frac{\sqrt{2\pi(d+k)}(\frac{d+k}{e})^{d+k}}{\sqrt{2\pi k}(\frac{k}{e})^k} e^{\frac{1}{12(d+k)} - \frac{1}{12k+1}} 2^{\frac{d^2(k-1)}{2k} - \frac{d(d+1)}{2}} \\ &\leq (d+k)^{d+k+\frac{1}{2}} 2^{\frac{d^2(k-1)}{2k} - \frac{d(d+1)}{2}} \\ &\leq (d+k)^{d+k+\frac{1}{2}} 2^{-\frac{d^2}{2k} + \frac{d}{2}} \\ &\leq \exp\Big((d+k+\frac{1}{2})\log(d+k) + \frac{d}{2}\log 2 - \frac{d^2}{2k}\log 2\Big).\end{aligned} \tag{53}$$

Since the d^2 term dominates, the limit will converge to 0. □

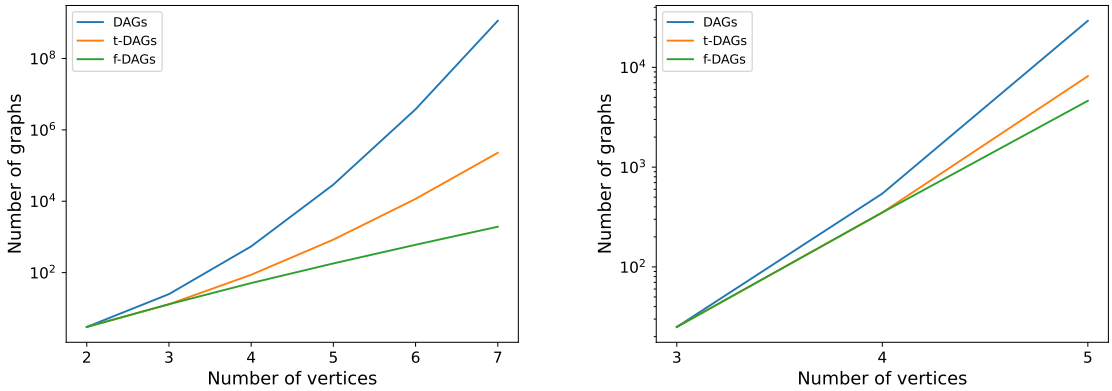


Figure 9: Comparison of the size of the sets of DAGs, t-DAGs, and f-DAGs as the number of vertices increases. We use two fixed values for the number of types: on the left $k = 2$ and on the right $k = 3$.

Proposition 10 (Limits of the cardinality, t-DAG vs f-DAG)**.** *The ratio* $|\mathcal{F}_d^k|/|\mathcal{G}_d^{k+1}|$ *converges to* 0 *when* $d \to \infty$ *and* k *is fixed.*

Proof. By Lopez et al. [2022], for a fixed ordering σ, $|\mathcal{F}_d^k(\sigma)| \leq k^d \cdot 2^{dk}$, since each of the d feature vertices can be assigned to one of k modules (k^d choices) and each feature vertex can connect to each of the k modules (2^{dk} edge configurations).

For the lower bound, by Proposition 9, for any single ordering σ:

$$|\mathcal{G}_d^{k+1}(\sigma)| \geq 2^{\frac{k\,d^2}{2(k+1)}} \tag{54}$$

since the uniform partition into $k+1$ groups of size $d/(k+1)$ yields $\frac{k\,d^2}{2(k+1)}$ possible inter-group edges.

Summing the f-DAG bound over all $d!$ orderings:

$$\frac{|\mathcal{F}_d^k|}{|\mathcal{G}_d^{k+1}|} \leq \frac{d! \cdot k^d \cdot 2^{dk}}{2^{\frac{k\,d^2}{2(k+1)}}}. \tag{55}$$

The numerator grows at most exponentially in $d \log d$ (since $d! \cdot k^d \cdot 2^{dk}$ is bounded by $e^{O(d \log d)}$), while the denominator grows as $2^{\Theta(d^2)}$. Since the d^2 term in the exponent of the denominator dominates, the ratio converges to 0 as $d \to \infty$. □

C.3 EMPIRICAL VALIDATION

In Fig. 9, we empirically validate some of the previous results. We generated all possible t-DAGs and f-DAGs (making sure to not include duplicates). It shows for $k = \{2, 3\}$ that $|\mathcal{F}_d^{k-1}| \leq |\mathcal{G}_d^k| \leq |\mathcal{G}_d|$. In particular, we can also observe the strict inequality $|\mathcal{F}_d^{k-1}| < |\mathcal{G}_d^k|$ when $d = k + 2$. It also corroborates that as d increases and k is fixed, the growth rate of DAGs, t-DAGs, and f-DAGs is of different orders.

D IDENTIFIABILITY

In this section, we provide an identifiability result based on the interplay of the graph and the types. We first restate a general data generating process for the graphs and then we prove the main theorem.

Definition 2 (Random sequence of growing t-DAG)**.** *We define a random sequence of t-DAGs* $(G_d^{T_d})_{d=0}^{\infty}$ *with* $G_d^{T_d} = (V_d, E_d)$ *and* $|V_d| = d$*, such that* $G_0^{T_0} = (\emptyset, \emptyset)$*. Each new t-DAG* $G_d^{T_d}$ *in the sequence is obtained from* $G_{d-1}^{T_{d-1}}$ *as follows: Create a new vertex* v_d *and sample its type* t *from a categorical distribution with probabilities* $p_1, ..., p_k$*. Let* $V_d = V_{d-1} \bigcup \{v_d\}$*. Let* $T_d(v_d) = t$ *and* $T_d(v_j) = T_{d-1}(v_j), \forall v_j \in V^{d-1}$*; To obtain* E_d*, for every vertex* $v_i \in V_{d-1}$*, add the edge* (v_i, v_d) *to* E_{d-1} *with probability* $p_{T_d(v_i), T_d(v_d)}$*.*

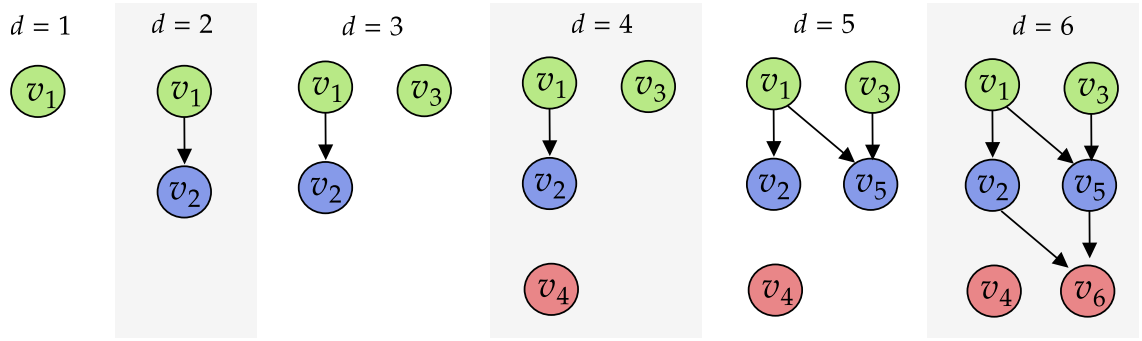


Figure 10: Example of a random sequence of growing t-DAGs. Assuming that there is 3 types (here denoted by colors), we can observe an oriented full-length path in the last step.

In Fig. 10 we show an example of a possible sequence of growing t-DAGs.

In the next section, we will first show that the probability of having a path of maximal length that is oriented in the MEC of the t-DAG grows exponentially fast to 1 as the graph grows with a fixed number of types. This allows us to find examples of metadata that belong to different types and also orient the minimal type DAG. With enough examples, it is then possible to learn a classifier that can infer the type of each vertex based on its metadata. From that, it is then possible to fully orient the rest of the graph.

D.1 THEORETICAL RESULTS

D.1.1 Oriented Paths in t-DAGs

We first remind the reader that, as proved by Verma and Pearl [1990], two DAGs are Markov equivalent if and only if they have the same skeleton and the same v-structures. The idea of the following proof is that many v-structures naturally occur in a t-DAG since two nodes of the same type can't cause each other.

Theorem 1 (Probability of oriented full-length path). *Let $(G_d^{T_d})_{d\geq 0}$ be a random sequence of growing t-DAGs with k types, type DAG of height ℓ, with $p_i \in (0,1)$ and $p_{ij} \in (0,1)$. Assume the type DAG has a single component. Let C be the indicator that the MEC of $G_d^{T_d}$ contains an oriented path of length ℓ. Then as d increases:*

$$P(C = 0 \mid d) \leq O(d) \cdot e^{-dr} \tag{17}$$

where $r = -\frac{1}{\ell}\ln(\alpha^*) > 0$ and $\alpha^* \in (0,1)$ depends only on $\{p_i\}$ and $\{p_{ij}\}$ (defined in the proof).

Proof. Without loss of generality, label the types $t_1, \ldots, t_\ell$ according to a topological ordering of the type DAG, so that t_1 is a root and edges between types go from lower to higher index. Fix ℓ and set $m := \lfloor d/\ell \rfloor$. We partition the ordered vertices into ℓ consecutive blocks of size m.

The proof proceeds in two stages. First, we orient a single edge using a v-structure at the root. Then, we extend this to a full-length path whose remaining edges are oriented by repeated application of Meek's first rule [Meek, 1995]: if $a \to b - c$ and a, c are not adjacent, then $b \to c$.

Stage 1: Orienting the first edge via a v-structure. We require block 1 to contain at least two vertices of type t_1, say v_{a_1} and $v_{a_1'}$, and block 2 to contain at least one vertex v_{a_2} of type t_2 that is a child of both v_{a_1} and $v_{a_1'}$. Since v_{a_1} and $v_{a_1'}$ share type t_1, no edge exists between them (Definition 1, property 2), so $v_{a_1} \to v_{a_2} \leftarrow v_{a_1'}$ is an unshielded collider. This orients the edge $v_{a_1} \to v_{a_2}$ in the MEC [Verma and Pearl, 1990].

Stage 2: Extending the path via Meek's rule. For each $i = 3, \ldots, \ell$, we require block i to contain at least one vertex v_{a_i} of type t_i satisfying two conditions:

1. v_{a_i} is a child of $v_{a_{i-1}}$ (to extend the chain), and
2. v_{a_i} is *not* adjacent to $v_{a_{i-2}}$ (to enable Meek's rule).

Given the already-oriented edge $v_{a_{i-2}} \to v_{a_{i-1}}$, condition 2 ensures that $v_{a_{i-2}} \to v_{a_{i-1}} - v_{a_i}$ with $v_{a_{i-2}}$ and v_{a_i} non-adjacent, so Meek's first rule forces the orientation $v_{a_{i-1}} \to v_{a_i}$.

By induction, all edges in the path $v_{a_1} \to v_{a_2} \to \cdots \to v_{a_\ell}$ are oriented in the MEC.

Events and failure probabilities. Define the following events on disjoint blocks:

- A_1: block 1 contains at least two vertices of type t_1.
- A_2: block 2 contains at least one vertex of type t_2 that is a child of both v_{a_1} and $v_{a_1'}$.
- A_i for $3 \le i \le \ell$: block i contains at least one vertex of type t_i that is a child of $v_{a_{i-1}}$ and is not adjacent to $v_{a_{i-2}}$.

For A_1: the number of type-t_1 vertices in block 1 is $\mathrm{Binomial}(m, p_1)$, so:

$$\Pr(\bar{A}_1) \le (1+m)(1-p_1)^{m-1}. \tag{56}$$

For A_2: each vertex independently fails to have type t_2 and be a child of both v_{a_1} and $v_{a_1'}$ with probability $1 - p_2 \cdot p_{1,2}^2$, so:

$$\Pr(\bar{A}_2 \mid A_1) \le (1 - p_2 \cdot p_{1,2}^2)^m. \tag{57}$$

For A_i with $3 \le i \le \ell$: each vertex independently succeeds with probability at least $p_i \cdot p_{(i-1),i} \cdot (1 - p_{(i-2),i})$, corresponding to receiving type t_i (probability p_i), being a child of $v_{a_{i-1}}$ (probability $p_{(i-1),i}$), and not being a child of $v_{a_{i-2}}$ (probability $1 - p_{(i-2),i}$). These are independent since type assignment and individual edge formations are independent in the growing t-DAG process. Thus:

$$\Pr(\bar{A}_i \mid A_1, \ldots, A_{i-1}) \le \big(1 - p_i \cdot p_{(i-1),i} \cdot (1 - p_{(i-2),i})\big)^m. \tag{58}$$

Union bound. Let

$$\alpha^* := \max\Big(1 - p_1,\ 1 - p_2 \cdot p_{1,2}^2,\ \max_{i \ge 3}\big(1 - p_i \cdot p_{(i-1),i} \cdot (1 - p_{(i-2),i})\big)\Big) \in (0,1). \tag{59}$$

Applying the union bound:

$$\Pr(C = 0 \mid d) \le (1+m)(1-p_1)^{m-1} + (\ell - 1)(\alpha^*)^m \le (\ell + m)(\alpha^*)^{m-1}. \tag{60}$$

Since ℓ is fixed and $m = \lfloor d/\ell \rfloor$, the prefactor $(\ell + m) = O(d)$ is polynomial while $(\alpha^*)^{m-1}$ decays exponentially:

$$\Pr(C = 0 \mid d) \le O(d) \cdot e^{-dr} \tag{61}$$

where $r := -\frac{1}{\ell}\ln(\alpha^*) > 0$. In particular, $\Pr(C = 1 \mid d) \to 1$ exponentially fast as $d \to \infty$. □

We note that this bound is conservative: in practice, convergence is faster because additional v-structures may form at intermediate steps of the chain, providing alternative orientations beyond those given by Meek's rule.

Remark 1 (Number of oriented full-length paths). The proof above establishes existence of at least one oriented path. In fact, the number of such paths grows polynomially. Let N denote the number of oriented full-length paths of length ℓ in the MEC. Any ordered ℓ-tuple $(v_{i_1}, \ldots, v_{i_\ell})$ independently forms an oriented path with some constant probability $c > 0$ (depending on p_i, p_{ij}, and the v-structure formation probability). Since there are $\binom{d}{\ell} = \Theta(d^\ell)$ such tuples, the expected number satisfies $\mathbb{E}[N] = \Omega(d^\ell)$. A standard second-moment argument (using the fact that the number of overlapping tuple pairs is $O(d^{2\ell-1})$ while $\mathbb{E}[N]^2 = \Omega(d^{2\ell})$) shows concentration around this mean. Thus, $N = \Theta(d^\ell)$ with high probability, providing abundant examples of each type in known orientations for learning the metadata-to-type mapping in Theorem 2.

D.1.2 Identifiability with Metadata

Using the previous result, we now show that using the metadata, we can identify the t-DAG.

Theorem 2 (Identifiability with metadata). *Assume that (i) the data-generating graph is a t-DAG with fixed k types, (ii) the metadata are i.i.d. conditioned on type and uniquely identify types, and (iii) the Markov equivalence class is recovered. Then, with probability approaching one as $d \to \infty$, the causal graph G is identifiable from the joint distribution $p(\boldsymbol{X}, \boldsymbol{M})$.*

Proof. We assume throughout that the observational distribution identifies the Markov equivalence class (MEC) of the data-generating DAG G, i.e., its skeleton and v-structures Verma and Pearl [1990], and that G is a t-DAG with a type DAG of height ℓ.

Step 1: Existence of an oriented full-length chain in the MEC. Consider the random sequence of growing t-DAGs as in Definition 2, where vertices are added sequentially with types drawn i.i.d. from a categorical distribution and edges are sampled according to type-dependent probabilities. By Theorem 1, assuming the type DAG has a single component and $p_i \in (0,1)$ and $p_{ij} \in (0,1)$, the probability that the MEC contains an *oriented* path of length ℓ converges to 1 exponentially fast as the number of vertices d grows.

Step 2: Using oriented chains to obtain labeled metadata samples. Assume the metadata M_v are i.i.d. conditional on the latent type (or, more weakly, on the semantic layer induced by the minimal type DAG). Collect vertices appearing on oriented full-length paths; by Theorem 1 and Remark 1, the number of such paths (and thus the number of collected vertices per layer) grows rapidly with d.

These vertices provide labeled examples $\{(M_v, \mathrm{layer}(v))\}$ for each semantic layer. Standard learning-theoretic arguments (PAC generalization) imply that, with sufficiently many i.i.d. labeled examples and a hypothesis class of reasonable complexity, one can learn a classifier $\hat{f}$ mapping metadata to layers with arbitrarily small error with high probability.

Step 3: Recovering layers and orienting all inter-layer edges. Apply $\hat{f}$ to all vertices to assign each vertex to a semantic layer. Given correct layer assignments, all edges between distinct layers are oriented by the layer order (from higher layer to lower layer, or equivalently along the oriented chain direction), yielding an oriented type-level partial order. Because G is type-consistent, this orientation agrees with the true direction between any two layers, and hence all inter-layer edges in the skeleton are oriented correctly.

Conclusion. Combining Steps 1–3, we recover (with probability approaching 1 as $d \to \infty$) the orientation of the full graph G from $p(\boldsymbol{X}, \boldsymbol{M})$. □

We note that, while this theorem shows that the graph can be recovered, the types themselves can only be recovered up to their semantic layers.

Remark 2 (Extension to multi-component type DAGs). The proof above assumes that the type DAG has a single component. When the type DAG consists of multiple connected components, the argument applies independently to each component. Specifically, Step 1 guarantees the existence of an oriented full-length path within each component, since Theorem 1 holds per-component under the same conditions $p_i \in (0,1)$ and $p_{ij} \in (0,1)$. Steps 2 and 3 then proceed component-wise: labeled metadata samples are collected separately for each component, a classifier $\hat{f}$ is learned for each component's semantic layers, and inter-layer edges within each component are oriented accordingly. Edges between variables belonging to different components are not constrained by the type ordering and remain oriented only to the extent permitted by the MEC. The conclusion therefore holds for each component independently, and the full graph G is recovered with probability approaching 1 as $d \to \infty$, provided each component receives a growing number of vertices.

D.1.3 Partial Identifiability under Metadata Coarsening

Finally, we discuss the case where the metadata can be less informative: the distributions of the metadata can be undistinguishable between different types.

Proposition 6 (Partial identifiability under coarsening)**.** *Identifiability holds at the quotient level: edges between variables whose aggregate types are distinguishable are oriented beyond the MEC, while edges within an aggregate class are oriented only to the extent permitted by v-structures. The quotient is well-defined as a DAG only when no type outside an equivalence class lies on a directed path between two types within it.*

Proof. Let G be the data-generating t-DAG with type assignment T into k types and type DAG $G_{\mathcal{T}}$ of height ℓ. Let $\sim$ be the equivalence relation with $t_i \sim t_j$ whenever the metadata distributions of t_i and t_j are indistinguishable. Write $[t]$ for the class of t and $\bar{T}(v) := [T(v)]$ for the induced *aggregate* type assignment, and let $\bar{G}_{\mathcal{T}} := G_{\mathcal{T}}/\!\sim$ be the quotient (Definition 9). We assume that $\bar{G}_{\mathcal{T}}$ is a valid quotient DAG (Definition 10) and, as in Theorem 2, that the observational distribution identifies the MEC of G.

Step 1 (Aggregate type-consistency). We first show that G is type-consistent with respect to $\bar{T}$ on edges between distinct classes. Suppose two distinct classes $[t_a] \neq [t_b]$ admitted edges in both directions, i.e. $u \to v$ and $u' \to v'$ with $\bar{T}(u) = \bar{T}(v') = [t_a]$ and $\bar{T}(v) = \bar{T}(u') = [t_b]$. Then $\bar{G}_{\mathcal{T}}$ would contain both $([t_a], [t_b])$ and $([t_b], [t_a])$, a 2-cycle, contradicting acyclicity. Hence all edges between any two distinct classes share a single direction, agreeing with the order of

$\bar{G}_{\mathcal{T}}$. Edges may still occur *within* a class, between variables whose fine types were merged; these are treated separately below.

Step 2 (Anchored ordering of the aggregate types). By Theorem 1, with probability approaching one the MEC of G contains oriented paths of length ℓ, and by Remark 1 the number of such paths is $\Theta(d^{\ell})$. Each oriented edge of such a path that joins two distinct classes fixes their relative order in the MEC, and by Step 1 this order is shared by all edges between those classes. As in the proof of Theorem 2, this abundance of oriented paths reveals the order of the quotient type DAG $\bar{G}_{\mathcal{T}}$ and supplies, for each aggregate semantic layer, a number of anchor vertices that grows with d. Since each aggregate class is a single node of $\bar{G}_{\mathcal{T}}$, it occupies a single semantic layer, so these anchors carry a well-defined aggregate-layer label.

Step 3 (Learning the metadata-to-class map and orienting inter-class edges). By definition of $\sim$, metadata are i.i.d. within a class and distinguishable across classes, hence uniquely identify the aggregate type (and, a fortiori, the aggregate layer). The anchors of Step 2 thus provide labeled samples $\{(M_v, \bar{T}(v))\}$ whose count grows with d, and standard PAC arguments yield a classifier $\hat{g}$ from metadata to aggregate types with vanishing error with high probability. Applying $\hat{g}$ to every vertex and orienting according to the recovered order of $\bar{G}_{\mathcal{T}}$ (Step 1) orients all edges between distinct aggregate classes correctly, beyond what the MEC alone determines. □

Remark 3 (Within-class edges and boundary cases). An edge between two variables of the same aggregate class joins fine types that are metadata-indistinguishable, so the metadata carries no signal for its direction. Such edges are oriented only to the extent permitted by the recovered skeleton and v-structures, i.e. up to the within-class MEC. In the extreme case where every type is its own class ($\sim$ is equality), then $\bar{G}_{\mathcal{T}} = G_{\mathcal{T}}$ and the statement reduces to Theorem 2. If all types are indistinguishable, the quotient collapses to a single node, there are no inter-class edges, and the procedure adds nothing to the MEC.

D.2 EMPIRICAL VALIDATION

In Fig. 12.a, we provide an empirical validation of our theoretical result. We use a random sequence of growing t-DAG, as defined in Def. 2, with $k = 3$, $p_i = \frac{1}{3}$ and $p_{ij} = 0.05$ for every i, j. We report the number of vertices of each type that are part of oriented full-length paths. It can be interpreted as the number of training examples that can be used to train a model on the metadata. For large graphs, there are more than 100 examples for each type.

We note that if p is relatively high, then the t-DAG becomes a single component with high probability. Similarly, for Erdős–Rényi graphs, it has been observed that when $p > 1/d$ the graph is a single component with high probability [Erdos et al., 1960]. We empirically show this phenomenon in Fig. 11. We note that when the graph is constituted of only one component, then it is identifiable even without knowing the types. V-structures are ubiquitously formed since vertices of the same type cannot be connected (in fact, for f-DAGs, the identifiability result of Lopez et al. [2022] relies on this fact). Besides a possible advantage in sample complexity, modelling with or without types leads to identifiability.

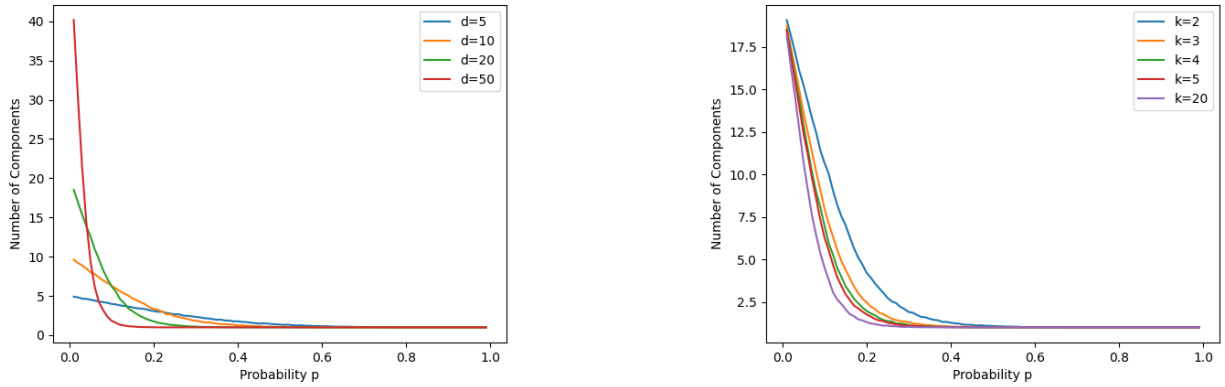
In Fig. 12.b, we show an example for $p = 0.02$. When d is not too high, this leads to graphs with several components. We compare the MEC size to the equivalence class when using types. We consider that the graph is identifiable if there are at least 10 examples of each type. We can see that for a regime with a low p, this identifiability result can be more useful than the classical MEC.

D.3 OTHER SOURCES OF IDENTIFIABILITY

Under stronger assumptions on the distribution of the metadata, causal graphs can be identified without even relying on our theoretical results.

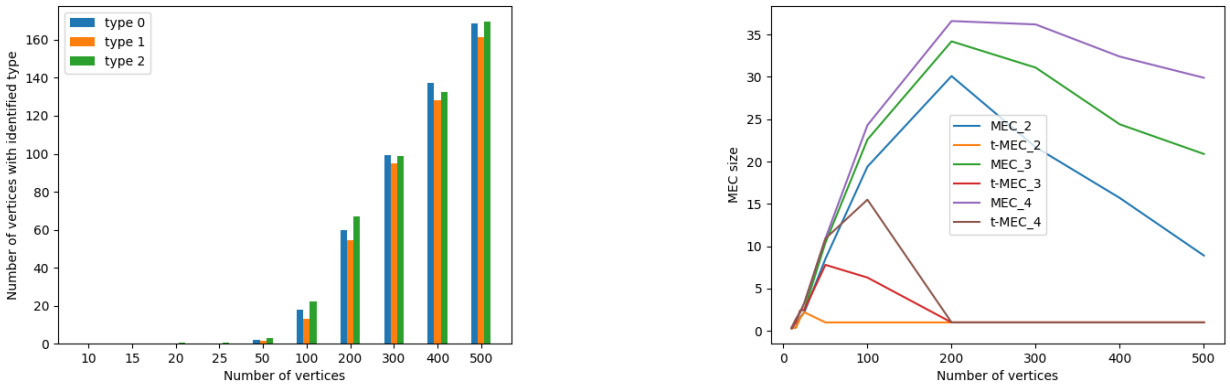
For example, if the metadata M follows a Gaussian mixture model (GMM), then we can identify the type of each variable since the components of GMM are identifiable up to permutation [McLachlan et al., 2019]. Furthermore, if g is an affine transformation, this result still holds since a GMM with affine transform remains a GMM [McLachlan et al., 2019]. Recovering the types leads us back to the previous results regarding the identifiability of the graph.

In general, we can also leverage identifiability results from the causal representation learning literature to recover the right latent space up to some benign transformations. As a specific case, if the metadata M follows a GMM transformed by a piecewise linear function f then we can leverage the identifiability results from Kivva et al. [2022] (See Theorems 3.9(c) and 3.10(d)). The latent space and the decoder function f are identifiable up to an affine transformation if the latent space is a GMM (even with an unknown number of components) and the function f is an injective piecewise affine function.



(a) (b)

Figure 11: Number of components in the graph w.r.t. the probability of adding edges. In a), the number of vertices d varies, and $k = 5$. In b), the number of types k varies and $d = 20$.



(a) (b)

Figure 12: In a), number of vertices of each type that are part of an oriented full-length path w.r.t. the total number of vertices (for a total of 3 different types). In b), comparison of the size of the equivalence class whether types are considered (t-MEC) or not (MEC) where the number $(2, 3, 4)$ correspond to the number of types.

E DATASET

E.1 SYNTHETIC DATASETS

All synthetic datasets are generated following this scheme: 1) the graph is sampled, 2) mechanisms are sampled and 3) data are sampled from the resulting model.

Graph Sampling. First, a type DAG $\mathbf{\Gamma} \in \{0, 1\}^{k \times k}$ is sampled following:

$$\mathbf{\Gamma} = \boldsymbol{P}^T \boldsymbol{U} \boldsymbol{P} \tag{62}$$

where $\boldsymbol{P} \in \{0, 1\}^{k \times k}$ is a permutation matrix randomly sampled and:

$$U_{ij} = \begin{cases} \text{Bern}(p) & \text{if } i < j \\ 0 & \text{otherwise.} \end{cases} \tag{63}$$

Then the type matrix $\boldsymbol{T} \in \{0, 1\}^{d \times k}$ is sampled for each variable (by rejection sampling we ensure that each type has at

least a variable associated to it):

$$\boldsymbol{T} \sim Cat(\boldsymbol{\pi}). \tag{64}$$

Finally, the skeleton $\boldsymbol{S} \in \{0,1\}^{d \times d}$ is sampled:

$$S_{ij} = \begin{cases} \text{Bern}(p_{\text{skel}}) & \text{if } i < j \\ S_{ji} & \text{otherwise.} \end{cases} \tag{65}$$

Together, these variable induce a graph $\boldsymbol{G} := \boldsymbol{S} \odot (\boldsymbol{T}^T \boldsymbol{\Gamma}\, \boldsymbol{T})$ that is, by construction, a consistent t-DAG. In our synthetic datasets, we use $p = 1, \boldsymbol{\pi} = \mathbf{1}, p_{skel} = 0.3$.

Mechanisms sampling. For each variable, we sample the parameters of the function f_i following the ordering of $\boldsymbol{G}$:

$$X_i = f(pa_i^{\boldsymbol{G}}(X), \epsilon_i; \theta_i). \tag{66}$$

For the linear function, we use:

$$X_i = \theta_i X + \epsilon_i \tag{67}$$

where the weight θ_{ij} for entries of $G_{ij} \neq 0$ are sampled as:

$$\theta_{ij} \sim \mathcal{U}([-1, -0.25] \cup [0.25, 1]). \tag{68}$$

For nonlinear functions, we use an additive noise model with neural networks:

$$X_i = \text{NN}(\boldsymbol{G}_{:i} \odot X; \theta_i) + \epsilon_i. \tag{69}$$

Data and Metadata Sampling. Finally, the data is sampled by ancestral sampling, i.e., each variable is sampled following the topological ordering of $\boldsymbol{G}$. For the synthetic datasets of the main experiments, we sample $n = 500$ examples. For the metadata, we use a Gaussian mixture model where each component corresponds to a type and $m = 2$.

E.2 PSEUDO-REAL DATASET

To generate the data, we used the same process as for the synthetic datasets (see App E.1) with the exception that we use the graph structure from Tu et al. [2019]. We generated 100 samples using linear mechanisms.

For the metadata, we first extracted the variable names and their category from Tu et al. [2019] and generated short descriptions using NotebookLM with the article and the GitHub repo as context. We show in Table 2 the codebook that is used for the pseudo-real dataset. Finally, to get embeddings from the descriptions, we use the `text-embedding-3-small` model via the OpenAI API [OpenAI, 2023]. It returns a 1536-dimensional embedding for each variable. We then use PCA to reduce the dimensionality to 50 and then t-SNE [Van der Maaten and Hinton, 2008] to further reduce it to 2 dimensions.

Table 2: Codebook for the Neuropathic Pain Dataset.

Variable Name	Category	Short Description
Craniocervical junction injury	Pathophysiological	Injury at the transition of the skull and neck.
Discoligamentous injury C1-C2	Pathophysiological	Injury to spinal disc or ligament at level C1-C2.
Discoligamentous injury C2-C3	Pathophysiological	Injury to spinal disc or ligament at level C2-C3.
Discoligamentous injury C3-C4	Pathophysiological	Injury to spinal disc or ligament at level C3-C4.
Discoligamentous injury C4-C5	Pathophysiological	Injury to spinal disc or ligament at level C4-C5.
Discoligamentous injury C5-C6	Pathophysiological	Injury to spinal disc or ligament at level C5-C6.
Discoligamentous injury C6-C7	Pathophysiological	Injury to spinal disc or ligament at level C6-C7.
Discoligamentous injury C7-C8	Pathophysiological	Injury to spinal disc or ligament at level C7-C8.
Discoligamentous injury C8-T1	Pathophysiological	Injury to spinal disc or ligament at level C8-T1.
Discoligamentous injury T1-T2	Pathophysiological	Injury to spinal disc or ligament at level T1-T2.
Discoligamentous injury T2-T3	Pathophysiological	Injury to spinal disc or ligament at level T2-T3.
Discoligamentous injury T3-T4	Pathophysiological	Injury to spinal disc or ligament at level T3-T4.
Discoligamentous injury T4-T5	Pathophysiological	Injury to spinal disc or ligament at level T4-T5.
Discoligamentous injury T5-T6	Pathophysiological	Injury to spinal disc or ligament at level T5-T6.
Discoligamentous injury T6-T7	Pathophysiological	Injury to spinal disc or ligament at level T6-T7.
Discoligamentous injury T7-T8	Pathophysiological	Injury to spinal disc or ligament at level T7-T8.
Discoligamentous injury T8-T9	Pathophysiological	Injury to spinal disc or ligament at level T8-T9.
Discoligamentous injury T9-T10	Pathophysiological	Injury to spinal disc or ligament at level T9-T10.

Variable Name	Category	Short Description
Discoligamentous injury T10-T11	Pathophysiological	Injury to spinal disc or ligament at level T10-T11.
Discoligamentous injury T11-T12	Pathophysiological	Injury to spinal disc or ligament at level T11-T12.
Discoligamentous injury T12-L1	Pathophysiological	Injury to spinal disc or ligament at level T12-L1.
Discoligamentous injury L1-L2	Pathophysiological	Injury to spinal disc or ligament at level L1-L2.
Discoligamentous injury L2-L3	Pathophysiological	Injury to spinal disc or ligament at level L2-L3.
Discoligamentous injury L3-L4	Pathophysiological	Injury to spinal disc or ligament at level L3-L4.
Discoligamentous injury L4-L5	Pathophysiological	Injury to spinal disc or ligament at level L4-L5.
Discoligamentous injury L5-S1	Pathophysiological	Injury to spinal disc or ligament at level L5-S1.
Discoligamentous injury S1-S2	Pathophysiological	Injury to spinal disc or ligament at level S1-S2.
Left C2 Radiculopathy	Pattern	Dysfunction of the left C2 nerve root.
Right C2 Radiculopathy	Pattern	Dysfunction of the right C2 nerve root.
Left C3 Radiculopathy	Pattern	Dysfunction of the left C3 nerve root.
Right C3 Radiculopathy	Pattern	Dysfunction of the right C3 nerve root.
Left C4 Radiculopathy	Pattern	Dysfunction of the left C4 nerve root.
Right C4 Radiculopathy	Pattern	Dysfunction of the right C4 nerve root.
Left C5 Radiculopathy	Pattern	Dysfunction of the left C5 nerve root.
Right C5 Radiculopathy	Pattern	Dysfunction of the right C5 nerve root.
Left C6 Radiculopathy	Pattern	Dysfunction of the left C6 nerve root.
Right C6 Radiculopathy	Pattern	Dysfunction of the right C6 nerve root.
Left C7 Radiculopathy	Pattern	Dysfunction of the left C7 nerve root.
Right C7 Radiculopathy	Pattern	Dysfunction of the right C7 nerve root.
Left C8 Radiculopathy	Pattern	Dysfunction of the left C8 nerve root.
Right C8 Radiculopathy	Pattern	Dysfunction of the right C8 nerve root.
Left T1 Radiculopathy	Pattern	Dysfunction of the left T1 nerve root.
Right T1 Radiculopathy	Pattern	Dysfunction of the right T1 nerve root.
Left T2 Radiculopathy	Pattern	Dysfunction of the left T2 nerve root.
Right T2 Radiculopathy	Pattern	Dysfunction of the right T2 nerve root.
Left T3 Radiculopathy	Pattern	Dysfunction of the left T3 nerve root.
Right T3 Radiculopathy	Pattern	Dysfunction of the right T3 nerve root.
Left T4 Radiculopathy	Pattern	Dysfunction of the left T4 nerve root.
Right T4 Radiculopathy	Pattern	Dysfunction of the right T4 nerve root.
Left T5 Radiculopathy	Pattern	Dysfunction of the left T5 nerve root.
Right T5 Radiculopathy	Pattern	Dysfunction of the right T5 nerve root.
Left T6 Radiculopathy	Pattern	Dysfunction of the left T6 nerve root.
Right T6 Radiculopathy	Pattern	Dysfunction of the right T6 nerve root.
Left T7 Radiculopathy	Pattern	Dysfunction of the left T7 nerve root.
Right T7 Radiculopathy	Pattern	Dysfunction of the right T7 nerve root.
Left T8 Radiculopathy	Pattern	Dysfunction of the left T8 nerve root.
Right T8 Radiculopathy	Pattern	Dysfunction of the right T8 nerve root.
Left T9 Radiculopathy	Pattern	Dysfunction of the left T9 nerve root.
Right T9 Radiculopathy	Pattern	Dysfunction of the right T9 nerve root.
Left T10 Radiculopathy	Pattern	Dysfunction of the left T10 nerve root.
Right T10 Radiculopathy	Pattern	Dysfunction of the right T10 nerve root.
Left T11 Radiculopathy	Pattern	Dysfunction of the left T11 nerve root.
Right T11 Radiculopathy	Pattern	Dysfunction of the right T11 nerve root.
Left T12 Radiculopathy	Pattern	Dysfunction of the left T12 nerve root.
Right T12 Radiculopathy	Pattern	Dysfunction of the right T12 nerve root.
Left L1 Radiculopathy	Pattern	Dysfunction of the left L1 nerve root.
Right L1 Radiculopathy	Pattern	Dysfunction of the right L1 nerve root.
Left L2 Radiculopathy	Pattern	Dysfunction of the left L2 nerve root.
Right L2 Radiculopathy	Pattern	Dysfunction of the right L2 nerve root.
Left L3 Radiculopathy	Pattern	Dysfunction of the left L3 nerve root.
Right L3 Radiculopathy	Pattern	Dysfunction of the right L3 nerve root.
Left L4 Radiculopathy	Pattern	Dysfunction of the left L4 nerve root.
Right L4 Radiculopathy	Pattern	Dysfunction of the right L4 nerve root.
Left L5 Radiculopathy	Pattern	Dysfunction of the left L5 nerve root.
Right L5 Radiculopathy	Pattern	Dysfunction of the right L5 nerve root.
Left S1 Radiculopathy	Pattern	Dysfunction of the left S1 nerve root.
Right S1 Radiculopathy	Pattern	Dysfunction of the right S1 nerve root.
Left S2 Radiculopathy	Pattern	Dysfunction of the left S2 nerve root.
Right S2 Radiculopathy	Pattern	Dysfunction of the right S2 nerve root.
Ibs	Symptom	Irritable bowel-like discomfort related to T10-L2 nerve roots.
Left neck problems	Symptom	Left-sided cervical discomfort.
Neck pain	Symptom	General pain in the cervical region.
Right neck	Symptom	Right-sided cervical discomfort.
Left tinnitus	Symptom	Ringing in the left ear.
Left eye problems	Symptom	Discomfort or visual issues with the left eye.
Left ear problems	Symptom	Discomfort in the left ear.
Right tinnitus	Symptom	Ringing in the right ear.
Right eye problems	Symptom	Discomfort or visual issues with the right eye.
Right ear problems	Symptom	Discomfort in the right ear.
Headache	Symptom	General cranial pain.
Left jaw problems	Symptom	Pain or dysfunction in the left jaw.
Left forehead headache	Symptom	Pain in the left frontal head region.
Mouth	Symptom	Oral discomfort.
Forehead headache	Symptom	Pain in the frontal head region.
Right headache	Symptom	Pain in the right head region.
Right pta	Symptom	Pain threshold assessment finding on the right side.

Variable Name	Category	Short Description
Pharyngeal discomfort	Symptom	Discomfort in the throat or pharynx.
Right jaw trouble	Symptom	Pain or dysfunction in the right jaw.
Back headache	Symptom	Pain at the posterior of the head.
Right back headache pain	Symptom	Localized pain in the right posterior head.
Left collarbone pain	Symptom	Discomfort in the left clavicle region.
Right collarbone problems	Symptom	Discomfort in the right clavicle region.
Central chest pain	Symptom	Midline thoracic pain.
Left central chest pain	Symptom	Left-midline thoracic pain.
Left central chest disorders	Symptom	Left-midline chest dysfunction.
Right front axle problems	Symptom	Right anterior shoulder or axillary pain.
Left shoulder impingement	Symptom	Left mechanical shoulder pain.
Right shoulder impingement	Symptom	Right mechanical shoulder pain.
Left shoulder problems	Symptom	General left shoulder dysfunction.
Left shoulder trouble	Symptom	Left shoulder discomfort.
Right shoulder problems	Symptom	General right shoulder dysfunction.
Right shoulder trouble	Symptom	Right shoulder discomfort.
Left upper arm discomfort	Symptom	Pain in the left brachial region.
Left upper elbow pain	Symptom	Pain above the left elbow joint.
Intracapular problems	Symptom	Pain between the shoulder blades.
Left interscapular complaints	Symptom	Discomfort between left scapula and spine.
Right intracapular trouble	Symptom	Discomfort between right scapula and spine.
Left lateral elbow pain	Symptom	Outer left elbow pain.
Left lateral arm discomfort	Symptom	Outer left arm pain.
Right lateral elbow pain	Symptom	Outer right elbow pain.
Left elbow problems	Symptom	General left elbow dysfunction.
Right elbow trouble	Symptom	General right elbow dysfunction.
Left arm	Symptom	Generalized pain in the left arm.
Left thumbs up	Symptom	Pain or trouble in the left thumb.
Right thumbs up	Symptom	Pain or trouble in the right thumb.
Left wrist problems	Symptom	Left wrist joint discomfort.
Right wrist problems	Symptom	Right wrist joint discomfort.
Left lower arm disorders	Symptom	Left forearm pain or dysfunction.
Right lower arm disorders	Symptom	Right forearm pain or dysfunction.
Left hand problems	Symptom	General left hand discomfort.
Right hand problems	Symptom	General right hand discomfort.
Left bend of arm problems	Symptom	Pain in the left antecubital fossa.
Right armband	Symptom	Tightness or pain around the right arm.
Right bend of arm discomfort	Symptom	Pain in the right antecubital fossa.
Left medial elbow problems	Symptom	Inner left elbow pain.
Right medial elbow problems	Symptom	Inner right elbow pain.
Left finger trouble	Symptom	Left digit pain or tingling.
Right finger trouble	Symptom	Right digit pain or tingling.
Left small finger trouble	Symptom	Left fifth digit discomfort.
Right little finger trouble	Symptom	Right fifth digit discomfort.
Left groin trouble	Symptom	Left inguinal region pain.
Left medial groin disorders	Symptom	Inner left groin pain.
Left lateral groin discomfort	Symptom	Outer left groin pain.
Central groin disorders	Symptom	Midline inguinal pain.
Right lateral groin discomfort	Symptom	Outer right groin pain.
Right groin trouble	Symptom	Right inguinal region pain.
Left adductor tendon	Symptom	Pain in the left inner thigh tendon.
Right adductor tendonitis	Symptom	Inflammation of the right inner thigh tendon.
Left hip disorders	Symptom	General left hip dysfunction.
Left backache	Symptom	Left lumbar or general back pain.
Backache	Symptom	General pain in the back.
Left lumbago	Symptom	Lower back pain on the left side.
Lumbago	Symptom	General lower back pain.
Right lumbago	Symptom	Lower back pain on the right side.
Left front thigh pain	Symptom	Anterior left thigh pain.
Right front thigh pain	Symptom	Anterior right thigh pain.
Right thigh problems	Symptom	General right thigh pain.
Left leg problems	Symptom	General left leg discomfort.
Left thigh pain	Symptom	General discomfort in the left thigh.
Right leg problems	Symptom	General right leg discomfort.
Right medial vadbesvär	Symptom	Inner right calf discomfort.
Left pta	Symptom	Pain threshold assessment finding on the left side.
Left hip joint	Symptom	Left hip joint-specific discomfort.
Right hip trouble	Symptom	Right hip discomfort.
Right hip arthritis	Symptom	Inflammatory-type pain in the right hip.
Left medial knee joint disorder	Symptom	Inner left knee joint pain.
Left front knee pain	Symptom	Anterior left knee discomfort.
Right medial knee joint disorder	Symptom	Inner right knee joint pain.
Right front knee pain	Symptom	Anterior right knee discomfort.
Left shin	Symptom	Anterior left lower leg pain.
Right shin	Symptom	Anterior right lower leg pain.
Left lower leg problems	Symptom	General left lower leg discomfort.
Left knee trouble	Symptom	General left knee joint pain.
Right knee trouble	Symptom	General right knee joint pain.

Variable Name	Category	Short Description
Left tåledbesvär	Symptom	Left toe joint complaints.
Left big toe problems	Symptom	Left first digit discomfort.
Right big toe problems	Symptom	Right first digit discomfort.
Left foot pain	Symptom	General left foot discomfort.
Left ankle trouble	Symptom	Left ankle joint pain.
Right ankle trouble	Symptom	Right ankle joint pain.
Left footstool trouble	Symptom	Pain in the dorsal left foot area.
Right arch	Symptom	Pain in the arch of the right foot.
Right morton trouble	Symptom	Nerve pain in the right toes.
Right fainting	Symptom	Episode of fainting or lightheadedness.
Left ischias	Symptom	Sciatic nerve pain on the left side.
Right ischias	Symptom	Sciatic nerve pain on the right side.
Left ham	Symptom	Left hamstring discomfort.
Left obesity	Symptom	Localized swelling or tissue issues on the left side.
Right ham	Symptom	Right hamstring discomfort.
Left toe problems	Symptom	General left toe discomfort.
Right foot pain	Symptom	General right foot discomfort.
Right tear problems	Symptom	Discomfort in the right lower extremity.
Right obesity	Symptom	Localized swelling or tissue issues on the right side.
Right dorsal knee joint disorder	Symptom	Posterior right knee pain.
Left dorsal knee joint disorder	Symptom	Posterior left knee pain.
Left lateral knee pain	Symptom	Outer left knee pain.
Right lateral knee pain	Symptom	Outer right knee pain.
Left small toe trouble	Symptom	Left fifth toe discomfort.
Left lateral foot disorders	Symptom	Outer left foot pain.
Right lateral foot disorders	Symptom	Outer right foot pain.
Right heel problems	Symptom	Right calcaneal discomfort.
Calcaneal pain	Symptom	General heel pain.
Left heel problems	Symptom	Left calcaneal discomfort.
Coccydyni	Symptom	Tailbone pain.
Left rear thigh pain	Symptom	Posterior left thigh pain.
Right rear thigh pain	Symptom	Posterior right thigh pain.
Left achilles problems	Symptom	Left Achilles tendon discomfort.
Left achilles tendon	Symptom	Left Achilles tendon pain.
Left achillodyni	Symptom	Specific left Achilles pain.
Right achilles problems	Symptom	Right Achilles tendon discomfort.
Right achilles tendency	Symptom	Sensitivity in the right Achilles tendon.
Right achillodyni	Symptom	Specific right Achilles pain.
Breast backache	Symptom	Pain in the thoracic back region.
Chest discomfort	Symptom	General thoracic region discomfort.
Left breast problems	Symptom	Left chest or breast area discomfort.
Right breast problems	Symptom	Right chest or breast area discomfort.
Toracal dysfunction	Symptom	Functional pain in the thoracic spine.
Upper abdominal discomfort	Symptom	Pain in the upper abdominal quadrants.
Lateral abdominal discomfort	Symptom	Pain in the side abdominal quadrants.
Abdominal discomfort	Symptom	General pain in the abdominal area.
Left lower abdominal discomfort	Symptom	Pain in the left lower abdominal quadrant.
Lower abdominal discomfort	Symptom	General lower abdominal pain.

F EXPERIMENTS

F.1 METRICS

$\mathbb{E}$-SHD. To compare a learned graph G to a ground-truth graph G^*, the Structural Hamming Distance (SHD) is commonly used. This corresponds to the total number of missing, superfluous and reversed edges. When using a Bayesian causal discovery method, we get a posterior and then use the expected SHD as in Lorch et al. [2021]:

$$\mathbb{E}\text{-SHD}(\boldsymbol{G}^*) := \mathbb{E}_{p(\boldsymbol{G}|D)}[\text{SHD}(\boldsymbol{G}, \boldsymbol{G}^*)] \tag{70}$$

I-NLL. For the interventional negative log-likelihood (I-NLL), we use:

$$\text{I-NLL} := \frac{1}{n_{\text{test}}} \sum_{k=1}^{n_{\text{test}}} \mathbb{E}_{p(\boldsymbol{G},\Theta|\mathcal{D})}[\text{NLL}(p^k, \mathcal{D}^k_{\text{test}})] \tag{71}$$

where

$$\text{NLL}(p, \mathcal{D}) := -\sum_{i=1}^{|\mathcal{D}|} \log p(\boldsymbol{x}^{(i)} \mid \Theta, \boldsymbol{G}). \tag{72}$$

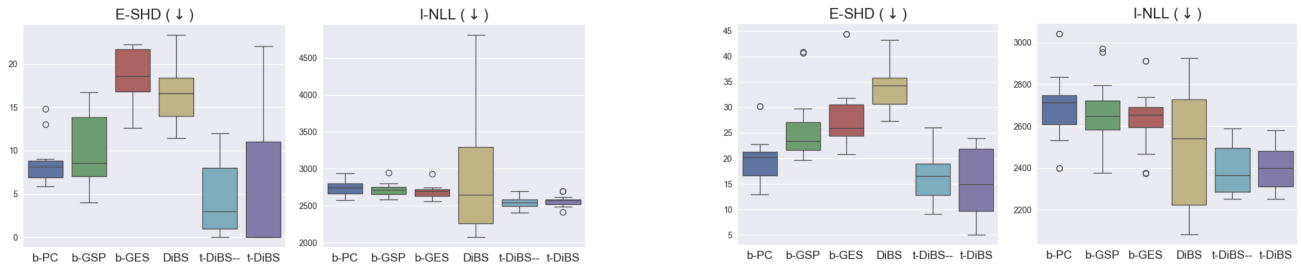
F.2 HYPERPARAMETERS

Following the DiBS work [Lorch et al., 2021], we use the gradient descent method RMSProp with a learning rate of 0.005. We use 20 particles, 128 samples for Monte Carlo estimations, minibatches of 100 examples, and we run the method for 5000 iterations. When applied on nonlinear data, we use MLPs with 2 hidden layers of 5 units. For the schedules, we use $\alpha_t := 0.1 \cdot t$ and $\gamma_t := 0.001 \cdot t$ where t is the number of iteration. For the sharpening prior, we use $\xi = 0.1$. For the bandwidths of the kernel, similar values to Lorch et al. [2021] were used, $\sigma_G = 5$, $\sigma_S = \{5, 15\}$ (for $d = \{20, 50\}$ respectively), and $\sigma_T = 0.01$. We upweight the metadata log-likelihood, $\log p(\mathbf{M} \mid \mathbf{T})$, by a factor of 500 in the joint score function. This reweighting was found empirically to improve performance, as it prevents the metadata signal from being overwhelmed by the larger observational likelihood. Finally, for the sparsity prior, we use $q = \frac{n \cdot d}{d(d-1)/2}$ with $n = \{1, 2\}$ for graphs of 50 and 20 variables, respectively.

Pseudo-real experiment. For t-DiBS and t-DiBS--, we only used one particle because of memory constraints. We did not include GES in the main results because the computation time exceeded 12 hours. We did not include DiBS since even after a hyperparameter search on the weight of the acyclicity constraint ($\beta = \{1, 10, 100\}$) and tests with different initialization schemes, the returned graphs were either cyclic or empty. For the PC method, we tested with $\alpha = \{0.05, 0.001, 0.0001\}$ and for GSP, $\alpha = \{0.01, 0.001, 0.0001\}$; we did not observe major changes for different values of these hyperparameters.

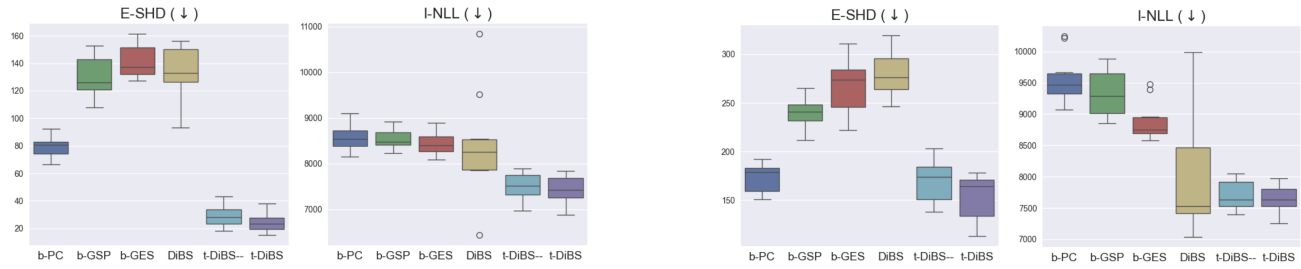
F.3 ADDITIONAL SYNTHETIC EXPERIMENTS

In this section, we report additional results on synthetic datasets. In Figure 13 and 14 we report performances on datasets with linear mechanisms with $d = \{20, 50\}$ variables and in Fig 15, datasets with nonlinear mechanisms with $d = 20$ variables.



(a) $k = 2$ types (b) $k = 5$ types

Figure 13: Performance on linear synthetic datasets with $d = 20$ variables and $k = \{2, 5\}$ types.

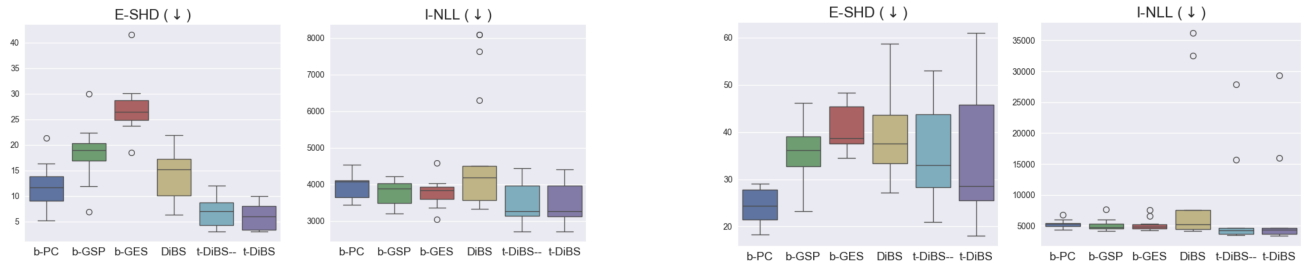


(a) $k = 2$ types (b) $k = 5$ types

Figure 14: Performance on linear synthetic datasets with $d = 50$ variables and $k = \{2, 5\}$ types.

F.4 ROBUSTNESS TO UNINFORMATIVE METADATA

To assess behavior when metadata provides no type signal, we apply t-DiBS to the (d=50, k=5) linear setting with metadata sampled from $\mathcal{N}(0, I)$ for all types. As shown in Table 3, performance degrades relative to informative metadata but remains comparable to type-agnostic baselines, confirming that t-DiBS recovers gracefully to MEC-level identifiability rather than failing catastrophically.



(a) $k = 2$ types

(b) $k = 5$ types

Figure 15: Performance on nonlinear synthetic datasets with $d = 20$ variables and $k = \{2, 5\}$ types.

Method	SHD
t-DiBS (uninformative metadata)	210.4 ± 22.9
t-DiBS (metadata)	154.0 ± 21.9

Table 3: Performance under uninformative metadata.

F.5 COMPARISON TO TYPED-PC

In this section, we evaluate t-DiBS against Typed-PC [Brouillard et al., 2022] on the synthetic experiments reported in the main text. We note here that Typed-PC assumes that ground-truth type assignments are known a priori. It should thus be considered as an upper bound on the performance since it has privileged access to the types. We modified Typed-PC by excluding intra-type edges, as is assumed in our setting. We use the majority variant (identified as best-performing in the original work) and use the Fisher and KCI conditional independence tests for linear and nonlinear settings, respectively. We use the implementation from `https://github.com/ServiceNow/typed-dag`.

We report the results in Fig 16. Typed-PC achieves better SHD overall, which is expected since it operates with access to ground-truth type information that t-DiBS must infer from metadata. Interestingly, the gap between the two methods is vanishing on the larger dataset, underscoring that t-DiBS recovers competitive structure despite operating under a strictly weaker assumption.

G METHODS

G.1 NONPARAMETRIC DAG BOOTSTRAP

Method	**Implementation**	**Link**
GSP	CausalDAG	`https://github.com/uhlerlab/causaldag`
PC	Causal-Learn	`https://github.com/py-why/causal-learn`
GES	pgmpy	`https://pgmpy.org/structure_estimator/ges.html`

Table 4: Causal discovery methods and their publicly available implementations.

We implement the nonparametric DAG bootstrap [Friedman et al., 1999] approach following the same procedure described in Agrawal et al. [2019] and also used in Lorch et al. [2021]. For each causal discovery method, we report in Table 4 the implementation used. Since these causal discovery methods output essential graph representing equivalence classes, we randomly sample a DAG for the learned class following the method described in Dor and Tarsi [1992]. To get a likelihood, we take the DAG outputted by the causal discovery methods and fit a model where each conditional is modeled by a neural network. For linear data, we use neural networks without hidden layer and for nonlinear data, we use 2 hidden layers of 5 units as used by the DiBS-like approaches. We train these models using gradient descent with the Adam optimizer.

For PC, we use the Fisher Z test and the KCI test [Zhang et al., 2011] respectively for linear and nonlinear data. Similarly,

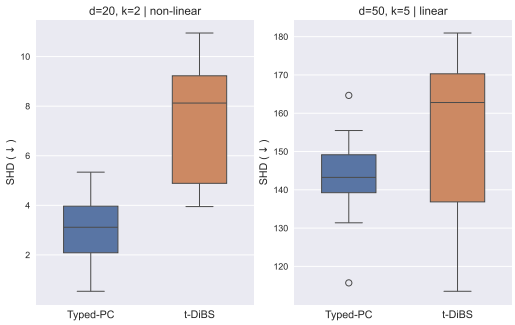


Figure 16: Comparison of t-DiBS toTyped-PC that has access to the ground-truth types.

for GSP, we use the partial correlation test and the KCI test [Zhang et al., 2011] respectively for linear and nonlinear data.